1

SCUOLA

Divisione Elettronica

olivetti

1

ELEA 4-115

Manuale di programmazione

MANUALE DI PROGRAMMAZIONE ELEA 4-115

edizione provvisoria

febbraio 1965

I N D I C E

	INTRODUZIONE	Pag. 1
CAP. 1	L'UNITA' CENTRALE	pag. 5
1.1 -	La memoria principale	pag. 5
1.2 -	Rappresentazione delle istruzioni in memoria	pag. 7
1.3 -	Rappresentazione dei dati in memoria	pag. 9
1.4 -	L'unità di governo	pag. 9
1.5 -	L'unità aritmetica	pag. 10
1.6 -	L'unità di governo entrata - uscita	pag. 11
1.7 -	Simultaneità operativa	pag. 14
CAP. 2	IL LINGUAGGIO SIMBOLICO APS	pag. 17
2.1 -	Generalità	pag. 17
2.2 -	Caratteristiche del linguaggio	pag. 17
2.3 -	Composizione minima dell'impianto	pag. 19
2.4 -	Formato delle frasi	pag. 19
2.5 -	Struttura delle frasi operative	pag. 20
2.6 -	Frasi dichiarative	pag. 25
2.7 -	Definizione di campi (DS)	pag. 25
2.8 -	Definizione di costanti (DC)	pag. 27
2.9 -	Definizione di comando esterno (DEC)	pag. 29
2.10 -	Inizio programma (START)	pag. 30
2.11 -	Fine programma (END)	pag. 30
2.12 -	Origine (ORG)	pag. 31
2.13 -	Salto a nuovo foglio (EJECT)	pag. 31
2.14 -	Salto di interlinee (SPACE)	pag. 32

CAP. 3	ISTRUZIONI DI TIPO ARITMETICO	pag. 33
3.1	- Generalità	pag. 33
3.2	- ADD : addizione decimale	pag. 34
3.3	- SOD : sottrazione decimale	pag. 38
3.4	- TFN : trasferimento numerico	pag. 41
3.5	- CFN : confronto numerico	pag. 43
3.6	- ADB : addizione binaria	pag. 45
3.7	- SOB : sottrazione binaria	pag. 47
CAP. 4	ISTRUZIONI DI TIPO LOGICO	pag. 51
4.1	- Generalità	pag. 51
4.2	- TFR : trasferimento	pag. 52
4.3	- CFR : confronto	pag. 54
4.4	- IMP : impaccamento	pag. 56
4.5	- DSP : disimpaccamento	pag. 57
4.6	- TCD : transcodifica	pag. 59
4.7	- RCA : ricerca avanti	pag. 62
4.8	- RCI : ricerca indietro	pag. 63
4.9	- ORL : Or	pag. 65
4.10	- AND : And	pag. 66
4.11	- ORE : Or esclusivo	pag. 68
4.12	- EDT : edizione	pag. 70
CAP. 5	ISTRUZIONI DI TIPO IMMEDIATO	pag. 77
5.1	- Generalità	pag. 77
5.2	- TFC : trasferimento di un carattere	pag. 77
5.3	- CFC : confronto di un carattere	pag. 78
CAP. 6	ISTRUZIONI DI SALTO	pag. 81
6.1	- Generalità	pag. 81
6.2	- SAC : salto su condizione	pag. 81
6.3	- SAR : salto e riserva indirizzo	pag. 83
6.4	- SAV : salti su condizioni di console od errore	pag. 84

CAP. 7		ISTRUZIONI VARIE	pag. 85
7.1	-	Generalità	pag. 85
7.2	-	ALT : istruzione di fermata	pag. 85
7.3	-	NOP : nessuna operazione	pag. 86
CAP. 8		ISTRUZIONE ESTERNA	pag. 87
8.1	-	Collegamento delle unità esterne	pag. 87
8.2	-	EST : istruzione esterna	pag. 88
8.2.1	-	Comando a unità esterna	pag. 89
8.2.2	-	Esame di condizione esterna	pag. 91
8.2.3	-	Trasferimento dati	pag. 93
CAP. 9		LA CONSOLE OPERATIVA	pag. 103
CAP. 10		NORME DA SEGUIRE NELLA COMPILAZIONE DEI PROGRAMMI	pag. 107

INTRODUZIONE

L'Elea 4-115 è un sistema elettronico per la elaborazione dei dati, di piccole dimensioni, dotato di elevata velocità di elaborazione, concepito per una gamma assai vasta di applicazioni.

Esso può essere dimensionato come impianto autonomo o come impianto ausiliario di elaboratori.

Come impianto autonomo il sistema Elea 4-115 è stato concepito per le applicazioni seguenti:

- sistema compatto a schede perforate: per le elaborazioni a schede perforate. Questo sistema è usato per sostituire o per ampliare centri meccanografici tradizionali, oltrechè per nuove installazioni;
- sistema ampliato con memorie magnetiche ausiliarie (nastri magnetici o memorie ad accesso casuale): inteso come sviluppo nel tempo dei sistemi precedenti, o per soddisfare le esigenze di utenti già orientati verso la utilizzazione di memorie magnetiche.

Come sistema ausiliario di altri elaboratori il sistema Elea 4-115 si presta alle seguenti applicazioni:

- stazione remota per trasmissioni a distanza, con eventuali funzioni di raccolta dati;
- sistema satellite collegato in linea a sistemi di maggior potenza di classe Elea o General Electric, con funzioni di calcolo o di governo unità in linea;
- sistema satellite fuori linea di elaboratori di classe Elea, General Electric, o di altre Case costruttrici, usato principalmente come convertitore schede perforate - - nastro magnetico - stampa e per altre applicazioni.

Il nucleo del sistema Elea 4-115 è l'Unità Centrale che, dal punto di vista funzionale, comprende:

- la memoria principale a nuclei magnetici;
- l'unità di governo;
- l'unità aritmetica;
- l'unità di governo di entrata - uscita;
- i governi di due unità integrate: un lettore seriale di schede o di documenti con caratteri magnetici ed una stampante parallela a caratteri normali o a caratteri magnetici.

Le caratteristiche di maggior rilievo della memoria principale sono:

- capacità da 2048 a 8192 caratteri di 8 bit utili,
- ciclo completo di memoria di 8 μs.,
- indirizzamento binario,
- controllo di parità.

Fra le caratteristiche principali dell'unità aritmetica ricordiamo:

- elaborazione di dati alfanumerici, decimali e binari;
- possibilità di elaborare dati di lunghezza variabile, mediante lunghezza espressa nelle istruzioni: fino a 16 caratteri per le operazioni aritmetiche, fino a 256 caratteri per le operazioni logiche e di trasferimento;
- istruzioni a due indirizzi;
- istruzioni logiche (AND, OR e OR esclusivo) per manipolazioni sul singolo bit;
- istruzione di edizione;
- istruzione di trascodifica.

Le operazioni su unità esterne sono controllate dalla unità di governo entrata-uscita le cui principali caratteristiche sono:

- due canali di entrata-uscita in "time-sharing" sulla memoria principale;
- capacità di collegare fino a 67 unità di entrata-uscita;
- trasferimento dei dati controllato dal programma;
- rilevazione e segnalazione automatica degli errori.

Con la unità centrale sono strettamente integrati due governi di unità esterne:

- governo per un lettore seriale di schede o documenti con caratteri magnetici;
- governo per una stampante parallela a caratteri normali o a caratteri magnetici. CMC 7

A tali governi si possono collegare le unità sottoelencate chiamate unità integrate:

- lettori seriali di schede perforate:
  - LS 300, velocità 300 schede-minuto,
  - LS 600, velocità 600 schede-minuto,
  - altri tipi di lettori seriali;
- stampanti parallele:
  - MZ 4 modello 1, velocità 300 linee-minuto,
  - MZ 4 modello 2, velocità 600 linee-minuto;
- unità per la lettura e la stampa di documenti con caratteri magnetici.

Sono poi collegabili al sistema attraverso l'interfaccia standard 4000, 65 unità, comunque scelte fra i seguenti tipi :

- le unità di introduzione e stampa già menzionate;
- stampante seriale MB, velocità 50 caratteri-secondo;
- perforatori di schede:
  - . P 85 di tipo seriale, velocità 100 colonne-secondo,
  - . P 150 di tipo parallelo, velocità 150 schede-minuto,
  - . P 301 di tipo parallelo, velocità 300 schede-minuto;
- lettore perforatore di schede :
  - . LP 300, velocità 300 schede-minuto
- lettore di nastro perforato :
  - . FL 400, velocità 400 caratteri-secondo;
- perforatore di nastro:
  - . PN 100, velocità 100 caratteri secondo;
- unità a nastro magnetico, con frequenza di trasferimento di :
  - . 15000 car. - sec.
  - . 20850 car. - sec.
  - . 30000 car. - sec.
  - . 41700 car. - sec.
  - . 60000 car. - sec.
- memorie magnetiche, ad accesso casuale.

byte = 8 bits

CAP. 1 L'UNITA' CENTRALE

1.1 - La memoria principale

La memoria principale dell'Elea 4-115 è costituita da nuclei di ferrite a ciclo di isteresi rettangolare.

Tali nuclei possono assumere due diverse condizioni di magnetizzazione alle quali sono convenzionalmente associati i numeri 0 e 1.

Si dirà di essere in presenza di "bit 1" quando un nucleo di ferrite si trova nella condizione di magnetizzazione indicata convenzionalmente con 1; in caso contrario si dirà di essere in presenza di "bit 0".

I nuclei di ferrite sono montati su 9 piani paralleli; 9 nuclei allineati su una stessa perpendicolare ai piani costituiscono una posizione di memoria.

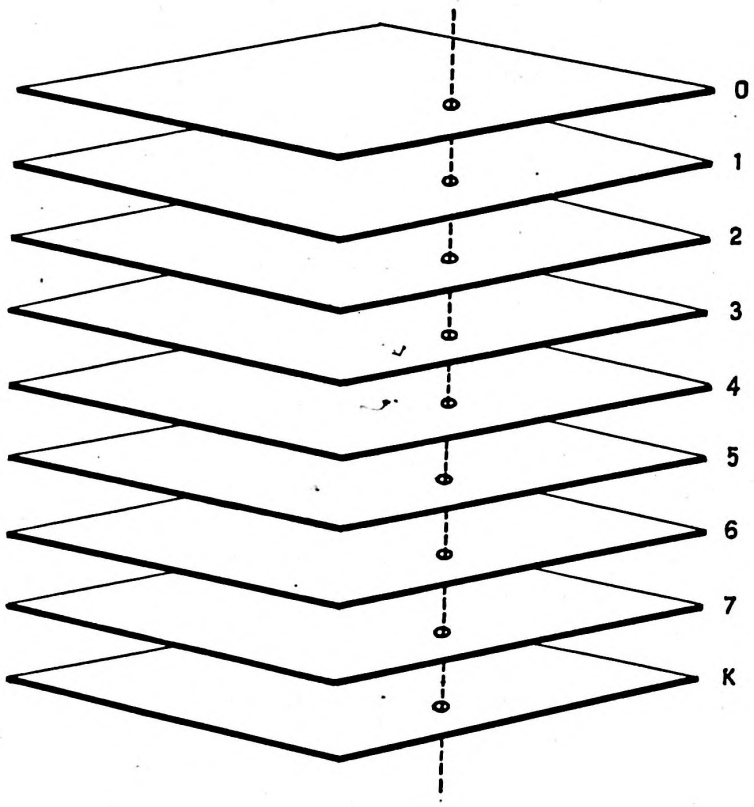
I nuclei dei primi 8 piani sono utilizzati per la registrazione dei dati e del programma, i nuclei del nono piano per il controllo di parità.

La memoria è dunque organizzata a parole di 8 bit utili, chiamate byte o caratteri.

Quando la configurazione di un byte presenta un numero pari di "bit 1", la macchina provvede a registrare il bit 1 di controllo.

Perciò la disparità del numero di "bit 1" di ogni carattere registrato in memoria, è condizione necessaria per l'esattezza della registrazione.

Se viene a mancare un "bit 1" o ne viene generato uno in una posizione in cui non avrebbe dovuto esservi, si avrà indicazione di errore, essendo pari il numero di "bit 1" della configurazione.



Rappresentazione di una posizione di memoria

La capacità della memoria può essere di 2048, 4096, 8192 byte. (caratteri)

Nel presente manuale ci si riferirà ad indirizzi relativi a tale estensione massima. Gli indirizzi sono espressi in codice binario puro mediante 13 bits, e variano perciò da 0 a 8191 occupando complessivamente due byte in memoria.

Per convenzione la memoria è rappresentata come una riga di caratteri con indirizzi crescenti da sinistra a destra.

Per i dati sia numerici che alfabetici la normale rappresentazione di macchina è quella effettuata disponendo il carattere più significativo a sinistra, come nella ordinaria scrittura.

1.2 - Rappresentazione delle istruzioni in memoria

Le istruzioni dell'Elea 4-115 sono a lunghezza variabile ed occupano da un minimo di due posizioni ad un massimo di sei posizioni di memoria.

Le configurazioni che le istruzioni possono assumere sono le seguenti:

		1	1	2	2
Formato 1	:	F	C	I1	I2
Formato 2	:	F	C	I1	
Formato 3	:	F	C	I2	
Formato 4	:	F	C		

(FUNZIONE)

- il carattere F è il codice di operazione; indica il tipo di operazione che deve essere eseguito dalla istruzione (per esempio somma, trasferimento, operazione di ingresso-uscita ecc.). I primi 2 bit di sinistra del carattere F, indicano il formato e quindi la lunghezza dell'istruzione, secondo i seguenti valori:

  . formato 1 : 11
  . formato 2 : 10
  . formato 3 : 01
  . formato 4 : 00

- il carattere C è utilizzato in dipendenza del codice di funzione per indicare:

  . lunghezza comune dei due operandi
  . lunghezza dei due operandi
  . un operando immediato
  . completamento del codice di funzione
  . nome di una unità di ingresso-uscita

- I1 ed I2 di due caratteri ciascuno, indicano rispettivamente l'indirizzo di memoria del primo e del secondo operando.

  I due operandi sono seguiti mediante due indirizzatori V1, V2, caricati con gli indirizzi I1 e I2 delle istruzioni.

1.3 - Rappresentazione dei dati in memoria

I dati registrati in memoria sono di lunghezza variabile. Le dimensioni degli operandi sono definite nelle istruzioni.

Ogni carattere o posizione di memoria è in grado di registrare un carattere alfanumerico o codice speciale nella rappresentazione convenzionale prescelta (ASCII, ISO, EBCDIC, etc.).

Qualunque sia il codice adottato, i dati numerici decimali vengono rappresentati utilizzando un byte per ogni cifra: i 4 bit meno significativi (quelli di destra detti "parte numerica") del byte rappresentano la cifra decimale in codice binario, i 4 bit più significativi (quelli a sinistra detti "zonatura") hanno un valore fisso e non sono interessati dalle operazioni aritmetiche decimali.

1.4 - L'unità di governo

La funzione dell'unità di governo è di prelevare dalla memoria principale le istruzioni, interpretarle e disporre i vari organi della macchina interessati alla esecuzione.

In seguito all'esame di ciascuna istruzione, l'unità di governo stabilisce il collegamento tra gli organi della macchina interessati e controlla l'esecuzione dell'operazione specificata nell'istruzione stessa (trasferimenti, operazioni logiche o aritmetiche, ecc.).

L'unità centrale comprende anche il quadro di comando, il quale dà la visualizzazione dello stato del sistema e dà inoltre la possibilità di accedere manualmente sulle operazioni del sistema.

1.5 - L'unità aritmetica

L'unità aritmetica esegue sia le operazioni aritmetiche binarie e decimali, che le operazioni di tipo logico.

Le operazioni aritmetiche decimali sono eseguite su numeri non segnati; ogni cifra deve trovarsi registrata in una posizione di memoria nei 4 bit meno significativi, mentre i rimanenti 4 bit della posizione sono ignorati.

La classificazione delle cifre decimali è la seguente:

0	=	0 0 0 0
1	=	0 0 0 1
2	=	0 0 1 0
3	=	0 0 1 1
4	=	0 1 0 0
5	=	0 1 0 1
6	=	0 1 1 0
7	=	0 1 1 1
8	=	1 0 0 0
9	=	1 0 0 1

Le operazioni aritmetiche binarie operano invece su dati espressi in binario puro, che impegnano tutti i bit delle posizioni di memoria nelle quali sono registrati. Esse servono principalmente per calcoli di indirizzi, espressi appunto in numeri binari.

Le istruzioni di tipo logico operano indifferentemente su dati alfabetici e numerici espressi in binario decimale o binario puro.

L'unità esegue le istruzioni sotto il controllo dell'unità di governo.

Alcune operazioni utilizzano ed aggiornano, in dipendenza del risultato ottenuto, il contenuto di due particolari organi di memoria aventi la capacità di un bit. Tali organi sono relativi alle condizioni RIPO (riporto) e DIVE (diverso da zero)

Le condizioni registrate possono essere consultate per stabilire una scelta di via nel seguito del programma, o per identificare ~~il segno algebrico di un risultato o di un confronto~~. condizioni di OVERFLOW, risultati in complemento o risultati di confronto.

Le diverse operazioni utilizzano in più modi queste segnalazioni, ed alcune in particolare eseguono un forzaggio iniziale, automatico, nella fase preparatoria.

Per ogni istruzione sarà specificato cosa avviene di RIPO e DIVE.

1.6 - L'unità di governo entrata-uscita

Una delle caratteristiche più importanti del sistema Elea 4-115 è l'unità di governo entrata-uscita.

L'unità di governo entrata-uscita comunica con le unità periferiche attraverso quattro connettori:

- il primo è specializzato per collegare direttamente, cioè senza il tramite del governo standard, una stampante parallela oppure, tramite governo standard, una stampante seriale;
- il secondo è specializzato per collegare direttamente, con le stesse modalità del primo connettore, un lettore

seriale di schede perforate o di documenti magnetici;

- il terzo presenta l'interfaccia standard 4000 e può essere utilizzato per collegare o una sola unità esterna dotata di governo standard, oppure fino a 64 unità esterne, tramite sincronizzatore e governi standard;
- il quarto (opzionale) presenta anch'esso l' interfaccia standard 4000 e permette il collegamento di una sola unità esterna dotata di governo standard.

Le funzioni di governo per la stampante parallela e per il lettore seriale, collegati ai primi due connettori, sono espletate da organi integrati nell'unità centrale, che vengono chiamati governi integrati. Per questa ragione le due unità suddette, quando sono collegate ai primi due connettori, vengono chiamate unità integrate.

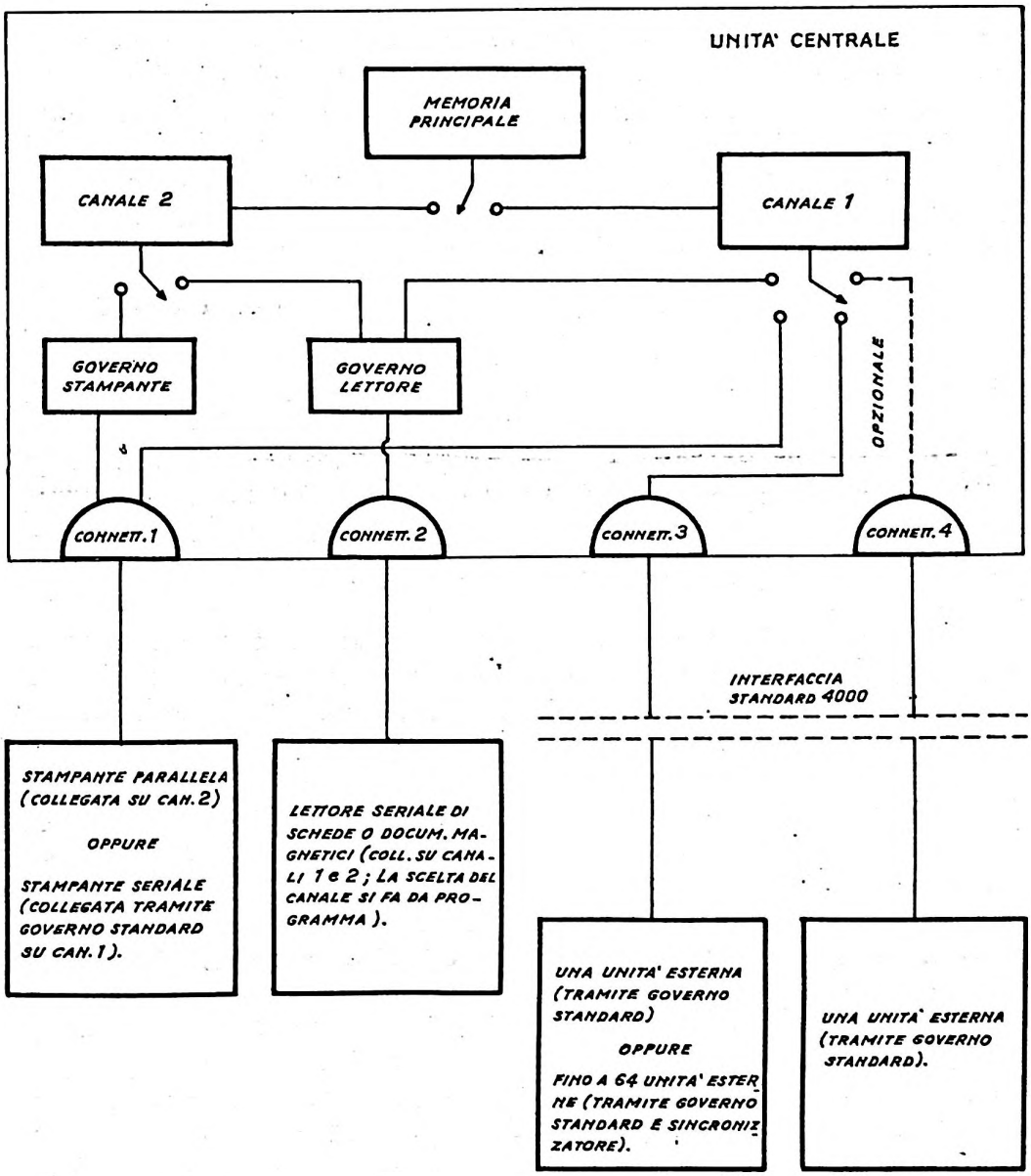
Le unità che si possono collegare al terzo ed al quarto connettore, attraverso l'interfaccia standard 4000, sono quelle elencate nel paragrafo introduzione.

Il numero di unità collegate ai quattro connettori, come si è visto, può arrivare fino a un massimo di 67.

Esse sono governate, per gli scambi di informazioni e di comandi da e verso la memoria principale, dall'unità di governo ingresso-uscita, che è costituita fondamentalmente da due canali in grado di operare contemporaneamente, in "time sharing" sulla memoria principale.

Il primo canale (canale 1) è utilizzato per il controllo della stampante seriale, del lettore integrato, e delle unità collegate al terzo ed al quarto connettore.

Il secondo canale (canale 2) è utilizzato per il controllo della stampante parallela integrata e del lettore seriale integrato.



Schema logico: la cirolazione delle informazioni

Da quanto detto risulta che il lettore seriale integrato può essere governato sia dal canale 1 che dal canale 2, tale scelta non è rigida in quanto viene fatta da programma.

Per chiarire quanto detto sui collegamenti delle unità esterne si riporta lo schema logico: la circolazione delle informazioni.

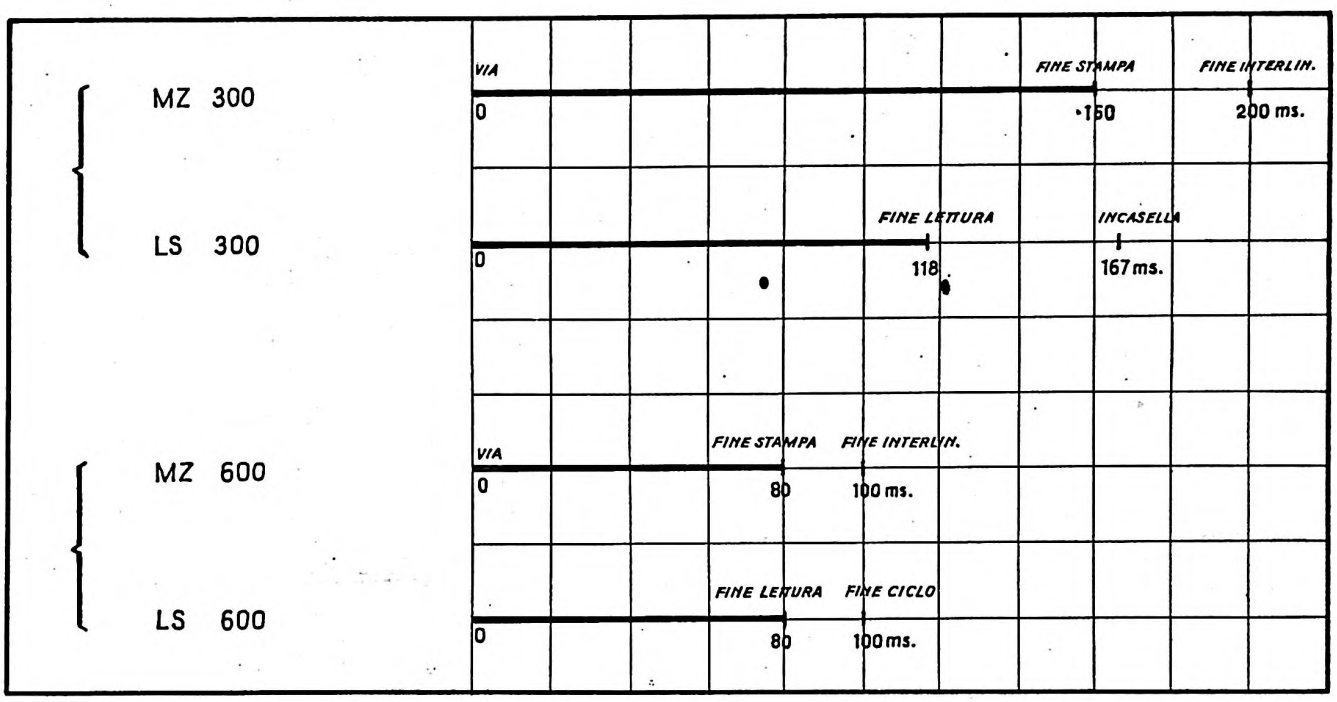
1.7 - Simultaneità operativa

Il sistema Elea 4-115 ha la caratteristica di poter sovrapporre operazioni di unità periferiche collegate rispettivamente ai due canali, attraverso un meccanismo di "time-sharing" sulla memoria principale.

E' possibile ad esempio sovrapporre i tempi operativi di una stampante collegata al canale 2 con una operazione di lettura o scrittura di qualunque elemento collegato al canale 1,

Inoltre i tempi puramente meccanici delle unità periferiche (come il salto della carta in una stampante o il ciclo di stampa in una stampante dotata di buffer) possono essere sovrapposti sia con le operazioni suddette che con l'esecuzione del programma da parte dell'unità centrale.

E' molto importante notare che sia il lettore che la stampante sono asincrone (cioè un ciclo può avere inizio a qualunque istante). Questa proprietà permette di evitare la perdita di un ciclo ogni qualvolta l'elaborazione sia più lunga del tempo disponibile; in realtà il tempo che si perde è solo la differenza tra il tempo di elaborazione e il tempo disponibile. Per esempio, nel caso di accoppiamento MZ 600 - LS 600 (il cui tempo disponibile è di 20 m.s.), se l'elaborazione dura 40 m.s., la velocità di lettura e stampa non si riduce a 300 ciclo-minuto, come avverrebbe se le due unità fossero sincronizzate, ma solo a 500 cicli-minuto.



Temporizzazioni unità esterne ELEA 4.115

CAP. 2 IL LINGUAGGIO SIMBOLICO APS

2.1 - Generalità

Il linguaggio APS 4-115 è un linguaggio simbolico di tipo uno a uno per la programmazione del sistema 4-115.

Con l'uso di questo linguaggio il programmatore può, pur senza rinunciare a nessuna delle caratteristiche di flessibilità offerte dal linguaggio base, descrivere in modo simbolico istruzioni di macchina, aree di memoria e costanti.

2.2 - Caratteristiche del linguaggio

Le principali caratteristiche dell'APS 4-115, che saranno più dettagliatamente descritte nel seguito, sono le seguenti:

- espressione dei codici operazione in maniera simbolica:

  codici di operazione simbolici, di facile apprendimento, sono previsti per tutte le istruzioni. Tali codici sono usati in luogo della configurazione binaria del codice di funzione, molto più complicata;

- codici simbolici estesi:

  sono previsti per le istruzioni di salto, codici simbolici estesi;

- riferimento in modo simbolico a indirizzi di memoria:

  alle istruzioni, aree di memoria e costanti, possono essere attribuiti nomi simbolici, ai quali il programmatore può fare riferimento, indipendentemente dall'indirizzo di memoria in cui saranno impiantati;

- facile rappresentazione dei dati:

  le costanti possono essere rappresentate dal programmatore in cifre decimali, caratteri alfanumerici, caratteri esadecimali e indirizzi di memoria. La conversione di tali dati nel linguaggio interno della macchina è uno dei compiti del compilatore APS 4-115;

- inserimento di sottoprogrammi:

  i sottoprogrammi di uso più frequente (come ad esempio i sottoprogrammi relativi alle operazioni di ingresso - - uscita) possono essere scritti e perforati su schede una sola volta. Una delle caratteristiche del compilatore è quella di poter inserire tali sottoprogrammi in qualunque programma. Il programmatore può fare riferimento ai sottoprogrammi in maniera simbolica;

- istruzioni per il compilatore:

  il linguaggio APS 4-115 comprende un gruppo di istruzioni per il compilatore, cioè istruzioni che non corrispondono ad istruzioni di macchina, ma forniscono informazioni utilizzate dal compilatore;

- stampa di programma:

  al termine di ogni compilazione il compilatore APS 4-115 fornisce su un'unica lista le istruzioni del linguaggio simbolico, con a fianco le corrispondenti istruzioni del programma oggetto;

- controlli di esattezza:

  le istruzioni del programma sorgente vengono controllate dal compilatore, per trovare eventuali errori dovuti ad un'errata utilizzazione del linguaggio. Gli errori trovati vengono segnalati nella lista del programma.

2.3 - Composizione minima dell'impianto

L'uso del compilatore APS 4-115 è previsto per macchine aventi almeno la seguente composizione:

- 4096 posizioni di memoria
- 1 lettore di schede
- 1 perforatore di schede
- 1 stampante (MZ o MB)

2.4 - Formato delle frasi

I programmi si compongono di frasi. Le frasi vengono distinte in frasi operative e frasi dichiarative:

- per frasi operative si intendono le frasi corrispondenti a istruzioni di macchina (somma, trasferimento, ecc.),
- per frasi dichiarative si intendono le frasi corrispondenti a istruzioni per il compilatore (esempio: definizione di aree o costanti).

Le frasi (operative o dichiarative) vengono scritte una per ogni riga di foglio di programmazione.

Ogni frase è costituita dai seguenti campi:

col.	1 - 4	codice identificazione
col.	32 - 35	codice sequenza
col.	36 - 41	nome della frase
col.	42 - 46	codice simbolico di istruzione
col.	47 - 74	operandi.

Il codice di identificazione è utilizzato per identificare il programma.

Il codice di sequenza contiene un numero di 4 cifre decimali delle quali le prime due indicano il numero di foglio nell'ambito del programma, le ultime due il numero di riga nell'ambito del foglio.

Il contenuto degli altri tre campi dipende dal tipo di frase e sarà descritto nei paragrafi seguenti.

2.5 - Struttura delle frasi operative

- nome:

  è usato per attribuire alla frase un nome simbolico di riferimento, la presenza del nome permette ad altre frasi di fare riferimento alla frase stessa.

  Se il nome esiste, la sua lunghezza può arrivare fino a 5 caratteri alfanumerici, il primo dei quali deve essere sempre alfabetico. All'interno del nome non sono ammessi caratteri spazio; la colonna 41 deve contenere sempre il carattere spazio.

- codice simbolico di istruzione:

  è usato per esprimere il codice simbolico dell'istruzione espresso nella frase. Il codice simbolico, la cui lunghezza varia da 1 a 5 caratteri, deve essere incolonnato a sinistra.

  Si riporta la lista dei codici simbolici delle istruzioni di macchina:

ADD	(AD)	addizione decimale	(add decimal)
SOD	(SD)	sottrazione decimale	(subtract decimal)
ADB	(AB)	addizione binaria	(add binary)
SOB	(SB)	sottrazione binaria	(subtract binary)
TFN	(MVD)	trasferimento numerico	(move decimal)
CFN	(CD)	confronto numerico	(compare decimal)
EDT	(EDT)	edizione	(editing)
TFR	(MV)	trasferimento	(move)
CFR	(C)	confronto	(compare)
IMP	(PK)	impaccamento	(pack)
DSP	(UPK)	disimpaccamento	(unpack)
ORL	(OR)	or	(or)
AND	(AND)	and	(and)
ORE	(EOR)	or esclusivo	(exclusive or)
TCD	(TR)	trascodifica	(translate)
RCA	(FSR)	ricerca avanti	(forward search)
RCI	(RSR)	ricerca indietro	(reverse search)
TFC	(MVI)	trasferimento carattere	(move immediate)
CFC	(CI)	confronta carattere	(compare immediate)
SAC	(BC)	salta su condizione	(branch on condition)
SAR	(BAS)	salta riservando indirizzo	(branch and store)
NOP	(NOP)	non opera	(no operation)
ALT	(HLT)	alt	(halt)
EST	(EXT)	operazione esterna	(external operation)

Oltre i codici suddetti (ciascuno dei quali corrisponde ad una istruzione di macchina) sono previsti, per comodità di programmazione, i seguenti codici simbolici estesi:

SNO	(NOB)	non salta	(no branch)
SAI	(B)	salta incondizionatamente	(branch inconditional)
SMG	(BH)	salta se maggiore	(branch on hig)
SMN	(BL)	salta se minore	(branch on low)
SEG	(BE)	salta se uguale	(branch egual)
SMGU	(BNL)	salta se maggiore o uguale	(branch on not low)
SMNU	(BNH)	salta se minore o uguale	(branch on not hig)
SDV	(BNE)	salta se diverso	(branch on not egual)
SCE1	(BS1)	salta se chiave 1	(branch on swicth 1)
SCE2	(BS2)	salta se chiave 2	(branch on swicth 2)
SEI	(BIE)	salta su errore interno	(branch on internal error)

- operandi:

il campo operandi è diviso in tanti sottocampi, quanti sono gli operandi richiesti dall'istruzione (uno o due). Ciascun sottocampo è di lunghezza variabile ed è separato dal successivo dal carattere virgola. Da questo punto di vista le istruzioni possono dividersi in cinque gruppi, come segue.

1° GRUPPO: nessun operando

ALT
NOP

2° GRUPPO: un operando (solo indirizzo)

SAI
SN 0
SMG
SMGU
SMN
SMNU
SEG
SDV
SAR
SCE1
SCE2
SEI

3° GRUPPO: due operandi (carattere e indirizzo)

SAC
TFC
CFC
EST

4° GRUPPO: due operandi (due indirizzi, con due lunghezze)

ADD
SOD
ADB
SOB
TFN
CFN

5° GRUPPO: due operandi (due indirizzi, con una lunghezza)

EDT
TFR
CFR
IMP
DSP
ORL
AND
ORE
TCD
RCA
RCI

Ciascun operando può essere costituito o da un carattere o da un indirizzo (con eventuale lunghezza).

Se si tratta di un carattere può essere espresso in uno dei modi seguenti:

- in esadecimale, incluso tra due apici, preceduto dal carattere X; ad esempio il carattere la cui configurazione binaria è 01011001 può essere espresso come X'59';
  di lunghezza sempre uguale a 2 caratteri
- in decimale, cioè dando l'equivalente decimale della configurazione binario del carattere; in questo caso non sono necessari gli apici, nè la lettera di contrassegno; il carattere dell'esempio precedente può essere espresso anche come 089.

- in alfanumerico, incluso tra due apici, preceduto dal carattere C; per esempio il carattere T viene espresso come C'T'.

Se si tratta di un indirizzo senza lunghezza può essere espresso:

- in decimale: dando l'equivalente decimale del valore binario dell'indirizzo, esempio: 4235,
- con un nome simbolico, esempio: MILA,
- con un nome simbolico aggiustato, cioè con un nome simbolico cui si somma o si sottrae una costante binaria, espressa in decimale, esempio: MILA+47, MILA-25.

Se si tratta di un indirizzo con lunghezza, l'operando viene espresso come nel caso precedente facendo però seguire la lunghezza scritta tra parentesi (in decimale); esempio: ~~MILA(5).~~ MILA(02), MILA+003(030)

L'indicazione di lunghezza può essere omessa nel caso di nomi simbolici aggiustati o no. In questo caso il compilatore assegna all'operando la stessa lunghezza assegnata al campo definito dal nome simbolico.

Il compilatore fa riferimento, nel caso di operandi espressi con nomi simbolici, all'indirizzo di destra dell'operando nel caso di istruzioni aritmetiche decimali o binarie, all'indirizzo di sinistra in tutti gli altri casi.

Un caso particolare di nome simbolico può essere considerato il simbolo * (asterisco); con questo simbolo si indica il valore attuale del contatore di memoria, cioè l'indirizzo che il compilatore assegna alla frase in corso di interpretazione.

2.6 - Frasi dichiarative

Le frasi dichiarative previste nel linguaggio APS 4-115 sono le seguenti:

DS	definizione di campo	(define storage)
DC	definizione di costanti	(define constant)
DEC	definizione di comando esterno	(define external command)
START	inizio programma	(start program)
END	fine programma	(end program)
ORG	origine	(set location counter)
EJECT	salto a nuovo foglio	
SPACE	salto di interlinee	

Nei paragrafi che seguono sarà descritto il significato e la struttura delle singole frasi dichiarative.

2.7 - Definizione di campi (DS)

La frase DS è utilizzata per riservare uno o più campi di elaborazione in memoria principale, assegnando al primo di essi un nome simbolico.

La struttura della frase DS è la seguente:

- nome: contiene il nome simbolico che si vuole attribuire al primo dei campi definiti dalla frase DS,
- codice di istruzione: DS,
- operandi: la struttura del campo operandi è la seguente:

dLn

dove: di lunghezza = a 3 caratteri

d (fattore di duplicazione) è un valore decimale che indica quanti campi devono essere riservati;

n indica la lunghezza in caratteri di ciascuno di essi: ed è sempre espresso mediante 3 caratteri decimali.
- se d=001, può essere omesso;
- n deve essere sempre espresso e deve essere diverso da 0;
- se si scrive d =000 non viene riservata alcuna area in memoria; tuttavia il nome del campo può essere utilizzato. Questa proprietà può essere sfruttata per definire sottocampi all'interno di un campo.

Nelle frasi operative in cui il nome del campo è espresso senza specificazione di lunghezza, si assume la lunghezza uguale a n; pertanto n deve essere non superiore di 256; esempi:

- 2 campi lunghi 80 caratteri:

```
C1    DS    002  L080
```

- 1 campo lungo 250 caratteri:

```
C2    DS    L250
```

- 1 campo lungo 20 caratteri e al suo interno 1 campo da 9, uno da 7 e uno da 4:

```
C1    DS    000  L010
C2    DS    L003
C3    DS    L007
C4    DS    L004
```

2.8 - Definizione di costanti (DC)

La frase DC è usata per generare una o più costanti in memoria. Il valore delle costanti da generare può essere espresso dal programmatore in caratteri alfanumerici, caratteri esadecimali, indirizzi assoluti o indirizzi simbolici.

La struttura della frase DC è la seguente:

- nome: contiene il nome simbolico che si vuole attribuire alla prima delle costanti generate dalla frase DC,
- codice di istruzione: DC,
- operandi: la struttura del campo operandi è la seguente:
  - dtLn'c' nel caso di costanti alfanumeriche o esadecimali,
  - ALn(c) nel caso di indirizzi, assoluti o simbolici,

  dove:

  ~~d è un valore decimale (fattore di duplicazione) che indica quante volte la costante deve essere ripetuta; se d=1 può essere omesso,~~

  t indica il tipo di costante e può assumere i seguenti valori:
  C = caratteri alfanumerici
  X = caratteri esadecimali
  A = indirizzo assoluto o simbolico (utilizzabile anche per esprimere in decimale qualsiasi valore da convertire in binario),

  Ln indica la lunghezza in byte della costante; tale lunghezza non può superare [illegible], 002 per il tipo A e 010 per i tipi C e X.

  c è il valore della costante, che deve essere compreso tra 2 apici nel caso di costanti alfanumeriche o esadecimali o tra parentesi tonde nel caso di indirizzi assoluti o simbolici. Il carattere apice ~~non può essere usato~~ all'interno delle costanti alfanumeriche, deve essere espresso raddoppiando due apici consecutivi.

La lunghezza implicita assegnata dal compilatore ad istruzioni che citano il nome della costante, semplicemente o con aggiustamento, è n.

Le costanti alfanumeriche vengono incolonnate sulla sinistra del loro campo; ~~le altre sulla destra.~~ le costanti esadecimali sulla destra.

Esempi:

- costanti alfanumeriche (C) :

. C1 DC ~~CL16'ROMA/NAPOLI'~~ CL010'ROMA'

produce in memoria
~~ROMA/NAPOLI/////~~ ROMA ××××LL

. ~~C2 DC 3CL3'ABC'~~

~~produce in memoria~~
~~ABCABCABC~~

. ~~C3 DC 3CL3'AB'~~

~~produce in memoria~~
~~AB¢AB¢AB¢~~

- costanti esadecimali:

. C1 DC ~~X'42FF'~~ XL002'42FF'

produce in memoria
42FF

. C2 DC ~~X'42'~~ XL002'42'

produce in memoria
0042

. ~~C3 DC 3XL2'42'~~

~~produce in memoria~~
~~004200420042~~

- indirizzi assoluti

  producono ~~sulla destra del campo definito~~, la costante, trasformata in binario,

- indirizzi simbolici

. C1 DC ~~A(PIPPO)~~ AL002(PIPPO)
produce sulla destra del campo C1, di 2 byte , il valore dell'indirizzo del campo PIPPO,

. C2 DC AL002~~A~~(PIPPO+20)
produce sulla destra del campo C2, di 2 byte , il valore dell'indirizzo del campo PIPPO aumentato di 20.

2.9 - Definizione di comando esterno (DEC)

La frase DEC è usata per generare in memoria un comando esterno di lunghezza due o sei byte.

La struttura della frase DEC è la seguente:

- nome: contiene il nome simbolico che si vuole attribuire al comando, cioè il nome da utilizzare altrove nel programma per riferirsi al comando stesso, (es.: operando nell'istruzione EST);

- codice istruzione: DEC;

- operandi: nel campo operandi si possono avere due operandi (per i comandi di due byte) o ~~quattro~~ 3 operandi (per i comandi di sei byte) separati da caratteri virgola:

  . il primo operando corrisponde al byte Z del comando e può essere espresso in esadecimale o in decimale, con le stesse modalità dell'operando carattere delle frasi operative,

- il secondo operando corrisponde al byte X e viene espresso con le stesse modalità del primo operando,
- ~~il terzo operando, se esiste, esprime la lunghezza del trasferimento; deve essere espresso in decimale,~~
- il ~~quarto~~ 3° operando, se esiste, è l'indirizzo del campo interessato al trasferimento dati. Esso può essere espresso con le stesse modalità delle frasi operative, cioè:
  - in decimale, dando l'equivalente decimale del valore binario dell'indirizzo,
  - con un nome simbolico,
  - con un nome simbolico aggiustato.

la lunghezza, se espressa, deve essere di 4 caratteri decimali. È chiusa fra parentesi e segue immediatamente l'indirizzo.

2.10 - Inizio programma (START)

La frase START è utilizzata per indicare l'inizio del programma da compilare. La sua struttura è la seguente:

- nome: contiene il nome del programma,
- codice istruzione: START,
- operandi: contiene l'indirizzo di impianto del programma, cioè il valore da dare inizialmente al contatore di memoria, ed il modulo di memoria per il programma oggetto e ~~per l'assemblatore~~. il nome del perforatore.

2.11 - Fine programma (END)

La frase END è utilizzata per indicare la fine del programma da compilare. La sua struttura è la seguente:

- nome: non è usato,
- codice istruzione: END,
- operandi: contiene un solo operando che esprime in decimale o con un nome simbolico (eventualmente con aggiustamento) l'indirizzo di lancio del programma.

2.12 - Origine (ORG)

La frase ORG è utilizzata per variare il contenuto del contatore di memoria. La sua struttura è la seguente:

- nome: non è usato,
- codice istruzione: ORG,
- operandi: contiene un solo operando che esprime il nuovo valore da introdurre nel contatore di memoria. Tale valore può essere espresso:
  - in decimale quando si vuol riportare il contatore di memoria ad un valore prefissato,
  - con il simbolo * +nnn (o * -nnn) quando si vuol aggiungere (o sottrarre) al contatore di memoria il valore n,
  - con il simbolo R che provoca l'arrotondamento su eccesso del valore attuale del contatore di memoria ad un multiplo di 256.

2.13 - Salto a nuovo foglio (EJECT)

La frase EJECT provoca, in fase di stampa del programma compilato, il salto a nuovo foglio prima della stampa della

frase successiva. La sua struttura è la seguente:

- nome: non è usato,
- codice istruzione: EJECT,
- operandi: non è usato.

2.14 - Salto di interlinee (SPACE)

La frase SPACE provoca, in fase di stampa del programma compilato, il salto del numero di interlinee indicato nel campo operandi, prima della stampa della frase successiva. La sua struttura è la seguente:

- nome: non è usato,
- codice istruzione: SPACE,
- operandi: contiene un solo operando che indica, in decimale il numero di interlinee da saltare.

MODULO DI PROGRAMMAZIONE ELEA 4-115

CODICE [1 | | | 4]

PAG. [32 | 33]

SISTEMA 4-115

RIGA 34 35	NOME 36 41	COD.OPERAZIONE 42 46	OPERANDI 47 50 55 60 65 70 74	NOTE
1 0				
1 5				
2 0				
2 5				
3 0				
3 5				
4 0				
4 5				
5 0				
5 5				
6 0				
6 5				
7 0				
7 5				
8 0				

CAP. 3 ISTRUZIONI DI TIPO ARITMETICO

3.1 - Generalità

L'istruzione è del tipo a 2 indirizzi. Questi indirizzi stanno ad indicare il carattere meno significativo dei campi di memoria che contengono i dati numerici da operare.

I campi sono denominati "campo 1" e "campo 2", gli indirizzi relativi I1 ed I2.

La fine dei campi viene definita per mezzo delle loro lunghezze "L1 e L2" espresse nell'istruzione stessa.

Il formato dell'istruzione che è a lunghezza fissa di 6 caratteri è la seguente:

F	L1 L2	I1	I2

- il primo carattere è il codice di funzione F. In esso i due bit più significativi hanno sempre il valore 11 ed i rimanenti 6 bit assumono il valore del codice vero e proprio;
- il secondo carattere esprime le lunghezze dei campi L1 ed L2. Esse sono uguali al numero di caratteri di ciascun campo diminuito di 1. Ad esempio la lunghezza di un campo di 9 caratteri è uguale a 8, quella di un campo di 1 carattere è 0.

  Poichè per esprimere una lunghezza sono disponibili 4 bit, la lunghezza di ogni campo può variare da 1 a 16 caratteri (L1 ed L2 variano da 0 a 15);

- il terzo e quarto carattere dell'istruzione stanno ad indicare l'inizio del campo 1: I1. Sono impegnati complessivamente gli 8 bit del quarto carattere ed i 5 bit meno significativi del terzo carattere. I 3 bit rimanenti devono essere zero.

  Gli indirizzi esprimibili variano dunque da 0000 a 8191;

- il quinto e sesto carattere indicano l'indirizzo del campo 2: I2. Le caratteristiche di questo indirizzo sono le stesse dell'I1.

Le operazioni di tipo aritmetico si svolgono su dati numerici in sistema decimale codificato binario o in sistema binario puro.

Se il dato è numerico decimale ogni cifra, espressa in binario, occupa i 4 bit meno significativi di una posizione di memoria. Se il dato è invece numerico binario tutti gli 8 bit delle posizioni di memoria che registrano il dato sono impegnati.

Le operazioni di tipo aritmetico si svolgono per indirizzi descrescenti.

3.2 - ADD : addizione decimale

Il campo 2 è sommato in modo decimale sul campo 1.

Il risultato appare nel campo 1.

CARATTERISTICHE - Ogni cifra di operando è registrata nei 4 bit meno significativi di posizione di memoria. I 4 bit più significativi (zonature) non intervengono nell'operazione.

Gli operandi possono essere lunghi da 1 a 16 caratteri e devono trovarsi incolonnati sulle posizioni decimali corrispondenti.

E' necessario che la lunghezza del campo 1 sia tale da contenere sempre il risultato.

ISTRUZIONE - F = 1111 1010

L1 = lunghezza campo 1
L2 = lunghezza campo 2
I1 = indirizzo campo 1
I2 = indirizzo campo 2

SVOLGIMENTO - L'operazione inizia dalle posizioni meno significative e procede quindi per indirizzi decrescenti. Se il campo 2 è più lungo del campo 1, ne viene considerata solo la parte meno significativa, per una lunghezza L1. Se il campo 2 è più breve del campo 1, l'operazione continua fino al termine di 1 prolungando fittiziamente 2, dalla parte più significativa con zeri.

Non viene sentita alcuna indicazione di segno.

All'inizio dell'operazione si dispongono automaticamente RIPO = 0 e DIVE = 0.

Durante lo svolgimento, per ogni coppia di caratteri sommati RIPO registra 1 se è stato generato il riporto, registra 0 in caso contrario. DIVE si posiziona a 1 se è generata una cifra di risultato diversa da 0 nel corso dell'operazione.

RISULTATO - Nel campo 1 l'operando è distrutto ed è registrato il valore della somma. L'operando del campo 2 è intatto.

Al termine dell'operazione la presenza di RIPO = 1 indica

condizione di over-flow (il risultato supera la capacità del campo).

La presenza di DIVE = 0 indica che il risultato generato è composto tutto di cifre uguali a zero.

I° ESEMPIO :

Esecuzione di una addizione decimale del campo MOV. sul campo TOT., essendo i campi rispettivamente di 3 e 4 caratteri.

ADD TOT. , MOV.

- campo MOV.

z7	z4	z5

- campo TOT. prima dell'operazione

z3	z6	z7	z2

- campo TOT. dopo l'operazione

z4	z4	z1	z7

II° ESEMPIO :

Esecuzione di una addizione decimale del campo C1 sul campo C2 essendo entrambi i campi di 3 caratteri.

ADD C2 , C1

- campo C1

z2	z4	z5

- campo C2 prima dell'operazione

z9	z8	z7

- campo C2 dopo l'operazione

z2	z3	z2

poichè il campo ricevente ha la lunghezza insufficiente a contenere il risultato, si ha segnalazione di overflow (RIPO = 1).

III° ESEMPIO :

Esecuzione di una addizione decimale del campo MARE su se stesso.

ADD MARE , MARE

- campo MARE prima dell'operazione

z2	z4	z7

- campo MARE dopo l'operazione

z4	z9	z4

si è raddoppiato il contenuto del campo.

3.3 - SOD : sottrazione decimale

Il campo 2 è sottratto in modo decimale dal campo 1.

Il risultato appare nel campo 1.

CARATTERISTICHE - Gli operandi sono trattati come entrambi positivi. Per le altre caratteristiche vale quanto detto per l'addizione.

ISTRUZIONE - F = 1111 1011

L1 = lunghezza campo 1
L2 = lunghezza campo 2
I1 = indirizzo campo 1
I2 = indirizzo campo 2

SVOLGIMENTO - All'inizio dell'operazione si dispongono automaticamente RIPO = 1 e DIVE = 0.

L'operazione inizia dalle posizioni meno significative e procede quindi per indirizzi decrescenti. Ogni cifra viene generata dalla somma del contenuto di RIPO con una cifra del campo 1 e la corrispondente del campo 2 complementata a 9. L'eventuale riporto creato in ogni somma si memorizza in RIPO e viene sfruttato nella somma successiva. Si ricorda che ad inizio operazione RIPO registra sempre 1.

DIVE si posiziona a 1 se nel corso dell'operazione si è generata una cifra di risultato diverso da zero.

Come nella somma, l'operazione termina quando è stato trattato l'ultimo carattere del campo 1.

RISULTATO - Nel campo 1 il minuendo è distrutto ed è registrato il risultato. Il sottraendo nel campo 2 è intatto.

Il significato finale della disposizione di RIPO e DIVE è il seguente:

- la presenza di RIPO = 1 indica che il risultato è positivo o nullo, e la presenza di DIVE = 0 in più indica che il risultato è nullo;
- la presenza di RIPO = 0 indica che il risultato è negativo e compare nel campo 1 in forma complementata a 10^{L1+1}.

I° ESEMPIO :

Esecuzione di una sottrazione decimale del campo C2 sul campo C1, essendo entrambi i campi di 4 caratteri.

SOD C1 , C2

- campo C2 (sottraendo)

z6	z2	z1	z8

- campo C1 (minuendo) prima della operazione

z0	z1	z8	z4

- campo C1 dopo l'operazione

z3	z9	z6	z6

essendo RIPO = 0 e DIVE = 1 il risultato è negativo e diverso da 0.

Inoltre è complementato a $10^{3+1} = 10.000$.

Volendo invece il valore assoluto del numero negativo risultante dall'operazione, bisogna sottrarre il risultato prima ottenuto, al campo 0000, non considerando pe

rò in questo caso la condizione RIPO = 0.

SOD C3 , C1 .

- campo C3

z0	z0	z0	z0

- il risultato ricomplementato, contenuto nel nuovo campo C3 dopo la operazione è pertanto:

z6	z0	z3	z4

II° ESEMPIO :

Esecuzione di una sottrazione decimale del campo ZERO su se stesso.

SOD ZERO , ZERO

- campo ZERO prima dell'operazione

z7	z4	z2	z3

- campo ZERO dopo l'operazione

z0	z0	z0	z0

si è azzerato il campo.

3.4 - TFN : trasferimento numerico

Il dato numerico del campo 2 è trasferito nel campo 1.

CARATTERISTICHE - Nel trasferimento sono interessati solamente i 4 bit meno significativi di ogni posizione di memoria. La lunghezza dei campi può essere al massimo di 16 caratteri.

ISTRUZIONE - F = 1111 1000

L1 = lunghezza campo 1
L2 = lunghezza campo 2
I1 = indirizzo campo 1
I2 = indirizzo campo 2

SVOLGIMENTO - L'operazione inizia dalle posizioni meno significative e procede quindi per indirizzi decrescenti.

L'operazione termina sempre su lunghezza del campo 1: se il campo 2 è più breve del campo 1, l'operazione continua fino al termine del campo 1 prolungando, fittiziamente, il campo 2 dalla parte più significativa con zeri.

All'inizio dell'operazione si dispongono automaticamente RIPO = 0 e DIVE = 0.

Se nel corso di un trasferimento si incontra un carattere diverso da 0, DIVE assume il valore 1.

RISULTATO - Nel campo 1 si trova registrato il dato numerico di 2 essendo rimaste intatte le zonature (4 bit più significativi).

La presenza di DIVE = 0 indica che il dato trasferito è composto tutto di cifre uguali a zero.

RIPO risulta sempre uguale a zero.

Il campo 2 rimane intatto.

I° ESEMPIO :

Esecuzione di un trasferimento numerico del campo MIO sul campo SUO, essendo entrambi i campi di 4 caratteri.

TFN SUO , MIO

- campo MIO

z8	z6	z0	z4

- campo SUO prima dell'operazione

z3	z6	z2	z7

- campo SUO dopo l'operazione

z8	z6	z0	z4

II° ESEMPIO :

Esecuzione di un trasferimento numerico del campo DEF sul campo ABC, essendo i campi rispettivamente di 3 e 5 caratteri.

TFN ABC , DEF

- campo DEF

z8	z9	z6

- campo ABC prima dell'operazione

z3	z2	z8	z9	z6

- campo ABC dopo l'operazione

z0	z0	z8	z9	z6

Si è voluto ottenere l'azzeramento delle posizioni a sinistra di un dato.

3.5 - CFN : confronto numerico

Il dato numerico del campo 1 viene confrontato con il dato numerico del campo 2.

Il risultato del confronto viene caricato in RIPO e DIVE. Esso può essere sfruttato dalle istruzioni di salto.

CARATTERISTICHE - L'operazione non tiene conto dei segni degli operandi che sono da considerare pertanto in valore assoluto.

Gli operandi, le cui cifre sono registrate nei 4 bit meno significativi di ogni posizione di memoria, possono essere lunghi da 1 a 16 caratteri e devono trovarsi incolonnati sulle posizioni decimali corrispondenti.

ISTRUZIONI - F = 1111 1001

L1 = lunghezza campo 1
L2 = lunghezza campo 2
I1 = indirizzo campo 1
I2 = indirizzo campo 2

SVOLGIMENTO - All'inizio dell'operazione si dispongono automaticamente RIPO = 1 e DIVE = 0.

Il confronto viene eseguito con l'operazione di sottrazione SOD senza registrazione del risultato nel campo 1.

RIPO e DIVE, come nella sottrazione, memorizzano le condizioni verificatesi.

RISULTATO - I segni del confronto sono materializzati dalle configurazioni di RIPO e DIVE:

- RIPO = 0 : 1 < 2
- RIPO = 1, DIVE = 0 : 1 = 2
- RIPO = 1, DIVE = 1 : 1 > 2
- RIPO = 0, DIVE = 0 configurazione impossibile

I campi 1 e 2 non sono modificati dall'operazione.

I° ESEMPIO :

Esecuzione di un confronto numerico del campo BETA con il campo ALFA, essendo i campi rispettivamente di 3 e 4 caratteri.

CFN ALFA , BETA

- campo BETA

z7	z9	z3

- campo ALFA

z6	z8	z4	z2

- risultato: RIPO = 1 DIVE = 1, quindi il campo ALFA è maggiore del campo BETA.

II° ESEMPIO :

Esecuzione di un confronto numerico del campo MARE con i 3 caratteri meno significativi del medesimo campo.

CFN MARE , MARE (3)

- campo MARE

z0	z0	z5	z9	z2	z7

- l'operazione termina lasciando invariato il campo con RIPO = 1 e DIVE = 1. Si osserva che se la parte significativa del dato registrato nel campo non avesse superato i tre caratteri, avremmo avuto uguaglianza tra i due termini, cioè RIPO = 1 e DIVE = 0.

  L'operazione risulta utile per riconoscere se il contenuto del campo supera un dato numero di cifre.

  L'operazione risulta utile per riconoscere il contenuto del campo supera un dato numero di cifre significative.

3.6 - ADB : addizione binaria

Il campo 2 è sommato in modo binario sul campo 1.

Il risultato appare nel campo 1.

CARATTERISTICHE - I dati da operare espressi in numero binario occupano tutti gli 8 bit delle posizioni di memoria che li registrano.

E' necessario che la lunghezza del campo 1 sia tale da contenere sempre il risultato.

ISTRUZIONE - F = 1111 1110

L1 = lunghezza campo 1
L2 = lunghezza campo 2
I1 = indirizzo campo 1
I2 = indirizzo campo 2

SVOLGIMENTO - L'operazione inizia dai bit meno significativi e procede per indirizzi decrescenti. Se il campo 2 è più lungo del campo 1, ne viene considerata solo la parte meno significativa, per una lunghezza L1.

Se il campo 2 è più breve del campo 1, l'operazione continuafino al termine di 1, prolungando fittiziamente 2 dalla parte meno significativa con caratteri costituiti da 8 bit zero.

Non viene sentita alcuna indicazione di segue.

All'inizio si dispongono automaticamente RIPO = 0 e DIVE = = 0.

Il riporto è generato secondo le regole dell'aritmetica binaria.

RISULTATO - Il valore della somma è contenuto nel campo 1. Il campo 2 è intatto.

DIVE viene disposto ad 1 se nel corso della somma si è generato un carattere di risultato contenente almeno un bit = = 1.

La presenza di RIPO = 1 indica condizione di over-flow.

ESEMPIO :

Esecuzione di una addizione binaria del campo C2 sul campo C1, essendo entrambi i campi di 2 caratteri.

ADB C1 , C2

- campo C2

0101 0111	0100 0101

(valore decimale = 22341)

- campo C1 prima dell'operazione

0010 0100	1100 0110

(valore decimale = 9414)

- campo C1 dopo l'operazione

0111 1100	0000 1011

(valore decimale = 31755)

3.7 - SOB : sottrazione binaria

Il campo 2 è sottratto in modo binario dal campo 1.

Il risultato appare nel campo 1.

CARATTERISTICHE - Gli operandi sono trattati come entrambi positivi. Le altre caratteristiche sono quelle dell'ADB.

ISTRUZIONI - F = 1111 1111.

L1 = lunghezza campo 1
L2 = lunghezza campo 2
I1 = indirizzo campo 1
I2 = indirizzo campo 2

SVOLGIMENTO - All'inizio dell'operazione si dispone automaticamente RIPO = 1 e DIVE = 0. L'operazione si svolge secondo le modalità dell'ADB con la differenza che i bit dei caratteri del campo 2 sono complementari a 1 prima di essere sommati sul campo 1.

RISULTATO - Il valore della differenza è contenuto nel campo 1.

La presenza di RIPO = 1 indica che il risultato è positivo o nullo, e la presenza di DIVE = 0 in più indica che il risultato è nullo.

Nel caso di RIPO = 0 il risultato è negativo, cioè è registrato in forma complementata a $_2 8$ (11 + 1).

ESEMPIO :

Esecuzione di una sottrazione binaria del campo PERA sul campo MELA, essendo i campi entrambi di 2 caratteri.

SOB MELA , PERA

- campo PERA

0010 0110	0101 0101

(valore decimale = 9.813)

- campo MELA prima dell'operazione

0100 0101	0111 1100

(valore decimale = 17.788)

- campo MELA dopo l'operazione

0001 1111	0010 0111

(valore decimale = 7.975)

- essendo il risultato positivo, sarà RIPO = 1, DIVE = 1.

CAP. 4 ISTRUZIONI DI TIPO LOGICO

4.1 - Generalità

L'istruzione è del tipo a 2 indirizzi. Questi indirizzi stanno ad indicare ~~per alcune istruzioni~~, la posizione più significativa degli operandi, ~~per altre istruzioni, la meno significativa.~~ eccetto che per l'istruzione di ricerca monetaria.

I campi che contengono gli operandi sono denominati campo 1 e campo 2, gli indirizzi relativi I1 e I2.

Per determinarela fine dei campi è sufficiente che l'istruzione, data la natura di queste operazioni, esprima una sola lunghezza denominata L.

Il formato dell'istruzione che è a lunghezza fissa di 6 caratteri è la seguente:

F	L	I1	I2

- il primo carattere è il codice di funzione F. In esso i due bit più significativi hanno sempre il valore 11 ed i rimanenti 6 bit assumono il valore del codice vero e proprio;
- il secondo carattere esprime la lunghezza L dei campi ai quali si riferisce. Si ricorda che per determinare la lunghezza di un campo si calcola la differenza fra gli indirizzi dei due caratteri, ESTREMI, ~~estremi,~~ che delimitano il campo ~~del quale si vuole conoscere~~ F. Poichè i bit disponibili per esprimere una lunghezza sono 8, L può variare da 0 a 255. Nelle operazioni di tipo logico la

lunghezza effettiva dei campi è quindi compresa fra 1 e 256 caratteri;

- i rimanenti 4 caratteri dell'istruzione stanno ad indicare gli indirizzi I1 ed I2. Ogni indirizzo occupa 2 caratteri e precisamente gli 8 bit del carattere meno significativo ed i 5 bit meno significativi del secondo carattere. I tre bit più significativi di quest'ultimo devono essere zero. Gli indirizzi esprimibili variano pertanto da 0000 a 8191.

Le operazioni di tipo logico interessano tutti gli 8 bit dei caratteri degli operandi e cioè trattano sia dati numerici che alfanumerici.

4.2 - TFR : trasferimento

Il contenuto del campo 2 è trasferito nel campo 1.

CARATTERISTICHE - I caratteri da trasferire possono essere di qualsiasi tipo (alfabetici, numerici, caratteri di programma aventi una qualsiasi configurazione binaria tra 0 e 255).

ISTRUZIONE - F = 1101 0010
L = lunghezza di entrambi i campi
I1 = indirizzo campo 1
I2 = indirizzo campo 2

SVOLGIMENTO - Il trasferimento, carattere per carattere, inizia dalla posizione più significativa e procede quindi per indirizzi crescenti.

Il numero dei caratteri da trasferire è indicato dalla lunghezza L espressa nella istruzione.

Le condizioni RIPO e DIVE non intervengono nella operazione.

RISULTATO - Il campo 1 viene azzerato e in esso viene registrato su tutti gli 8 bit il campo 2.

Il campo 2 non viene modificato.

Le condizioni RIPO e DIVE posizionate precedentemente allo svolgimento di questa istruzione permangono immutate.

ESEMPIO :

Esecuzione di un trasferimento del campo ROMA sul campo LODI essendo i campi entrambi di 6 caratteri.

TFR LODI , ROMA

- campo ROMA

G	I	U	L	I	O

- campo LODI prima della operazione

U	G	O	z0	z4	z6

- campo LODI dopo l'operazione

G	I	U	L	I	O

4.3 - CFR : confronto

Il contenuto del campo 1 viene confrontato con il contenuto del campo 2. Il risultato del confronto viene caricato in RIPO e DIVE. Esso viene sfruttato dalle istruzioni di salto.

CARATTERISTICHE - I caratteri da confrontare possono essere di qualsiasi tipo (alfabetici, numerici, caratteri di programma aventi una qualsiasi configurazione binaria tra 0 e 255).

ISTRUZIONE - F = 1101 0101
L = lunghezza di entrambi i campi
I1 = indirizzo campo 1
I2 = indirizzo campo 2

SVOLGIMENTO - Il confronto, carattere per carattere, inizia dalla posizione più significativa dei campi e procede per indirizzi crescenti arrestandosi sulla lunghezza L indicata nella istruzione.

Ad inizio operazione si dispongono automaticamente RIPO = 1 e DIVE = 0.

L'operazione, come per la sottrazione binaria, si svolge sommando ogni carattere di 2, complementato bit per bit, con il riporto ed i bit del corrispondente carattere del campo 1. Il riporto generato da ogni singola somma interviene nella somma successiva. Il risultato numerico della operazione non viene registrato. L'operazione termina non appena DIVE sia disposto a 1, segno che il risultato ha qualche bit eguale a 1 oppure dopo aver confrontato L + 1 caratteri.
RIPO e DIVE memorizzano le condizioni verificatesi.

RISULTATO - I segni del confronto sono materializzati dalle configurazioni di RIPO e DIVE:

- RIPO = 0 : 1 < 2
- RIPO = 1, DIVE = 0 : 1 = 2
- RIPO = 1, DIVE = 1 : 1 > 2

Se il confronto avviene su dati alfanumerici è necessario, per avere il risultato maggiore o minore, che il loro ordine di peso sia coerente con l'ordine dei numeri binari puri rappresentati dai corrispondenti codici. Diversamente la condizione posta ha solo il significato di uguale se DIVE = 0 o di diverso se DIVE = 1.

I campi 1 e 2 non sono modificati dall'operazione.

ESEMPIO :

Esecuzione di un confronto del campo VINO con il campo SALE essendo i campi entrambi di 2 caratteri.

CFR SALE , VINO

- campo VINO

1001 0010	0000 0100

- campo SALE

1001 0010	1000 0010

- Essendo il contenuto del campo SALE > VINO si avrà RIPO = 1 e DIVE = 1.

  Il contenuto dei campi sarà peraltro immutato.

4.4 - IMP : impaccamento

L'operazione consiste nel creare con le modalità esposte in seguito, 1 carattere nel campo 1 partendo da 2 caratteri del campo 2.

ISTRUZIONE - F = 1101 1010
L = lunghezza campo 1
I1 = indirizzo campo 1
I2 = indirizzo campo 2

SVOLGIMENTO - L'operazione interessa alternativamente una coppia di caratteri del campo 2 ed un carattere del campo 1, procedendo nel verso degli indirizzi crescenti. I 4 bit meno significativi del primo e del secondo carattere di 2 sono trasferiti nei 4 bit più e nei 4 bit meno significativi del primo carattere del campo 1.

Il trasferimento si ripete in modo analogo per i caratteri successivi e si arresta sulla lunghezza L espressa nell' istruzione, la quale si riferisce al campo 1.

Le condizioni RIPO e DIVE non intervengono nell'operazione.

RISULTATO - Il campo 1 immagazzina il dato numerico secondo le modalità esposte. Il campo 2 rimane intatto.

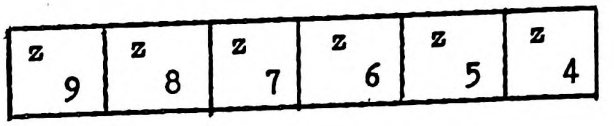
Le condizioni RIPO e DIVE posizionate precedentemente allo svolgimento di questa istruzione permangono immutate.

ESEMPIO :

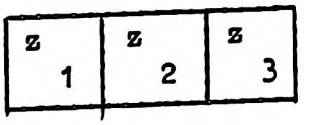
Esecuzione di un impaccamento del campo BRERA sul campo DUCA, essendo il campo DUCA di 3 caratteri:

IMP DUCA , BRERA

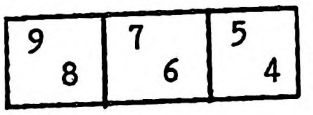
- campo BRERA



- campo DUCA prima dell'operazione



- contenuto del campo DUCA dopo l'operazione



4.5 - DSP : disimpaccamento

Partendo da un carattere del campo 2 si creano, con le modalità sotto esposte, due configurazioni nei 4 bit meno significativi di due caratteri del campo 1.

ISTRUZIONE - F = 1101 1000
L = lunghezza campo 2
I1 = indirizzo campo 1
I2 = indirizzo campo 2

SVOLGIMENTO - L'operazione interessa alternativamente un carattere del campo 2 e due caratteri del campo 1, procedendo nel verso degli indirizzi crescenti.

I 4 bit più ed i 4 bit meno significativi del carattere di 2 sono inseriti nei 4 bit meno significativi dei due caratteri successivi di 1.

L'operazione termina sulla lunghezza L dell'istruzione la quale si riferisce al campo 2.

RISULTATO - Il dato trattato è registrato in modo normale nel campo 1 ed è conservato impaccato nel campo 2.

Poichè l'operazione interessa solo i 4 bit meno significativi di 1, le zone di questo campo restano immutate.

Come nell'istruzione di impaccamento le condizioni RIPO e DIVE rimangono immutate.

ESEMPIO :

Esecuzione di un impaccamento dal campo DUCA sul campo BRERA, essendo il campo DUCA di 3 caratteri:

```
DSP     BRERA , DUCA
```

- campo DUCA

9 8	7 6	5 4

- campo BRERA, prima dell'operazione

z 1	z 2	z 3	z 3	z 2	z 1

- contenuto del campo BRERA dopo l'operazione

z 9	z 8	z 7	z 6	z 5	z 4

4.6 - TCD : transcodifica

Permette di tradurre una informazione espressa con un dato codice, in un altro codice.

CARATTERISTICHE - Il campo 2 ~~registra~~ contiene la tabella di transcodifica. Per indicare l'indirizzo di tale campo si utilizzano solamente i 5 bit più significativi di I2 mentre i rimanenti 8 bit devono essere nulli.

L'indirizzo del campo 2 può quindi assumere una delle seguenti configurazioni:

I2 =	0000	0000	0000	0000	= 0
	0000	0001	0000	0000	= 256
	0000	0010	0000	0000	= 512
	0000	0011	0000	0000	= 768
	⋮	⋮	⋮	⋮	⋮
	0001	1111	0000	0000	= 7936

Come si nota, all'indirizzo I2 non si può assegnare un valore che non sia multiplo di 256.

I caratteri della tabella di trascodifica devono essere registrati nel campo 2 agli indirizzi relativi, contenuti negli 8 bit meno significativi dell'indirizzo di 2, definiti dalla espressione binaria dei caratteri da transcodificare.

Esempio: Si supponga di calcolare l'indirizzo del carattere "%" di una certa tabella di transcodifica registrata con origine I2 = 0000 0010 0000 0000 (512) sapendo che il carattere "%" deve essere ~~scritto ha la configurazione 1011 1100 (188)~~. posto all'indirizzo 700

I2 -	0000	0010	0000	0000	
% -			1011	1100	
Ind. car% della tabella -	0000	0010	1011	1100	(700)

Il carattere "%" della tabella, non può trovarsi che all'indirizzo di memoria 700.

Il campo 1 registra i dati che sono da transcodificare.

ISTRUZIONE - F = 1101 1100
L = lunghezza campo 1
I1 = indirizzo campo 1
I2 = indirizzo campo 2

SVOLGIMENTO - Ogni carattere del campo 1 è inserito al posto degli 8 bit meno significativi dell'indirizzatore del campo 2. All'indirizzo ottenuto si legge un carattere che rimpiazza il carattere di 1.

L'operazione si svolge per indirizzi crescenti e termina sulla lunghezza L del campo 1.

RISULTATO - Nel campo 1 sono registrati i dati espressi secondo il nuovo codice.

Il campo 2 conserva la tabella di codificazione.

Le condizioni RIPO e DIVE non interessate dall'operazione, rimangono immutate.

ESEMPIO :

Esecuzione di una transcodifica del campo LORO con una tabella di passaggio di codifica da sistema EBCDIC a sistema Iso, registrata nel campo NOSTRO, essendo il campo LORO di 8 caratteri.

TCD LORO , NOSTRO

- campo NOSTRO, con tabella da EBCDIC a ISO

tabella da EBCDIC A ISO

- campo LORO, prima dell'operazione, con espressione in caratteri EBCDIC

z2	z2	Ø	M	A	R	Z	O

- campo LORO, dopo l'operazione, con espressione in caratteri ISO

z2	z2	Ø	M	A	R	Z	O

4.7 - RCA : ricerca avanti

Dato un carattere da considerare come termine campione, la istruzione, seguendo il verso degli indirizzi crescenti, ricerca nella zona di memoria contenente un dato, un carattere uguale al carattere campione, dando come risultato l'indirizzo del carattere seguente.

CARATTERISTICHE - Il carattere campione è di tipo qualsiasi e si trova nel campo 2 costituito da una sola posizione. La ricerca avviene per indirizzi crescenti nel campo 1.

ISTRUZIONE - F = 1101 1001
L = lunghezza campo 1
I1 = indirizzo campo 1
I2 = indirizzo campo 2

SVOLGIMENTO - All'inizio dell'operazione vengono poste automaticamente le condizioni RIPO = 1 e DIVE = 0.
Qualora la ricerca ha avuto esito positivo DIVE viene posto a 1

RISULTATO - E' un indirizzo, che viene depositato agli indirizzi 254 e 255 della memoria.

Si ottiene l'indirizzo del carattere successivo a quello trovato, se la ricerca ha avuto esito positivo, ~~o all'ultimo carattere di destra di 1, se la ricerca ha dato esito negativo~~ o l'indirizzo successivo a quello di fine Tabella se la ricerca ha dato esito negativo.

ESEMPIO :

Esecuzione di una ricerca avanti con campo GEL0 di 9 caratteri e campo NEVE contenente il carattere campione.

RCA GEL0 , NEVE

- campo NEVE

W

- campo GEL0

0	P	E	R	A	Ø	W	z3	z6

- il carattere W ricercato è all'interno del campo GEL0 per cui risulta che il suo indirizzo aumentato di 1 si registra nelle posizioni 254 e 255 della memoria.

  DIVE risulta uguale a uno.

4.8 - RCI : ricerca indietro

Come nella RCA, dato un carattere da considerare come termine campione, l'istruzione, seguendo il verso degli indirizzi decrescenti, ricerca nella zona di memoria contenente un dato, l'indirizzo del carattere uguale al carattere campione, dando come risultato l'indirizzo del carattere precedente in memoria.

CARATTERISTICHE - Sono le stesse della RCA con la differenza che la ricerca nel campo 1 avviene per indirizzi decrescenti.

ISTRUZIONE - F = 1101 1011
L = lunghezza campo 1
I1 = indirizzo campo 1
I2 = indirizzo campo 2

SVOLGIMENTO - All'inizio dell'operazione vengono poste automaticamente le condizioni RIPO = 1 e DIVE = 0.
L'operazione procede per indirizzi decrescenti del campo 1.
Se nel corso dell'esplorazione del campo 1 si trova un carattere uguale al carattere posto nel campo 2, l'operazione termina e risulta DIVE = 1.
Viceversa, se non si fosse trovato tale carattere nel campo 1, si avrebbe DIVE = 0.

RISULTATO - E' un indirizzo che viene depositato agli indirizzi 254 e 255 della memoria.

Il risultato è l'indirizzo del carattere trovato, diminuito di 1, se la ricerca ha avuto esito positivo, o l'indirizzo dell'estrema sinistra della tabella diminuita di 1, se la ricerca ha avuto esito negativo.

ESEMPIO :

Esecuzione di una ricerca indietro con campo ROSSO di 10 caratteri e campo ROSA contenente il carattere campione.

RCI ROSSO , ROSA

- campo ROSA

+

- campo ROSSO

z6	+	z1	z3	z2	z7	+	z8	z2	z6

- Il primo carattere + è identificato nel campo ROSSO e il suo indirizzo diminuito di 1 si registra nelle posizioni 254 e 255 della memoria.
  DIVE risulta uguale a uno.

4.9 - ORL : Or

Esegue la somma logica del campo 2 con il campo 1.
Il risultato appare nel campo 1.

CARATTERISTICHE - I dati da operare possono essere quantità binarie qualsiasi.

ISTRUZIONE - F = 1101 0110
L = lunghezza di entrambi i campi
I1 = indirizzo campo 1
I2 = indirizzo campo 2

SVOLGIMENTO - L'operazione procede nel verso degli indirizzi crescenti. Di ogni coppia di caratteri dei due campi vengono combinati i bit corrispondenti.
I bit del risultato vengono ottenuti secondo la legge seguente:

bit di 1	bit di 2	risultato
0	0	0
0	1	1
1	0	1
1	1	1

RISULTATO - Il campo 1 contiene il risultato della somma logica.
Le condizioni di salto permangono immutate.

ESEMPIO :

Esecuzione di una ORL del campo TRE col campo DUE, essendo i campi di 2 caratteri.

ORL DUE , TRE

- campo TRE	0011 0000	0000 1111
- campo DUE, prima della operazione	0101 1101	0110 0110
- campo DUE, dopo l'operazione	0111 1101	0110 1111

- Si volevano portare ad 1 il 3°, 4° e gli ultimi 4 bit del campo DUE.

4.10 - AND : And

Esegue il prodotto logico del campo 2 con il campo 1.
Il risultato appare nel campo 1.

CARATTERISTICHE - I dati da operare possono essere quantità binarie qualsiasi.

ISTRUZIONE - F = 1101 0100
L = lunghezza di entrambi i campi
I1 = indirizzo campo 1
I2 = indirizzo campo 2

SVOLGIMENTO - L'operazione procede nel verso degli indirizzi crescenti.
Di ogni coppia di caratteri dei due campi vengono combinati i bit corrispondenti.
I bit del risultato vengono ottenuti secondo la legge seguente:

bit di 1	bit di 2	risultato
0	0	0
0	1	0
1	0	0
1	1	1

Le condizioni RIPO e DIVE non intervengono nell'operazione.

RISULTATO - Il campo 1 contiene il risultato del prodotto logico.
Le condizioni di salto permangono immutate.

ESEMPIO :

Esecuzione di una AND del campo TRE col campo DUE essendo i campi di 2 caratteri.

AND DUE , TRE

- campo TRE	0011 1111	1111 0000
- campo DUE, prima della operazione	0101 1101	0110 0111
- campo DUE, dopo l'operazione	0001 1101	0110 0000

- Si volevano portare a 0 il 1°, il 2° e gli ultimi 4 bit del campo DUE.

4.11 - ORE : Or esclusivo

Esegue la somma logica esclusiva del campo 2 con il campo 1. Il risultato appare nel campo 1.

CARATTERISTICHE - I dati da operare possono essere quantità binarie qualsiasi.

ISTRUZIONE - F = 1101 0111
L = lunghezza di entrambi i campi
I1 = indirizzo campo 1
I2 = indirizzo campo 2

SVOLGIMENTO - All'inizio della operazione vengono poste le condizioni RIPO = 1 e DIVE = 0.

I bit del risultato vengono ottenuti secondo la legge seguente:

bit di 1	bit di 2	risultato
0	0	0
0	1	1
1	0	1
1	1	0

Se nel corso dell'operazione viene generato almeno 1 bit di risultato uguale ad 1, si pone DIVE = 1.

RISULTATO - Il campo 1 contiene il risultato della somma logica esclusiva.
RIPO è sempre 1. DIVE = 1 indica un risultato diverso da zero.

ESEMPIO :

Esecuzione di una ORE del campo TRE col campo DUE.

	ORE	DUE	TRE
- campo TRE		0011 0000	0000 1111
- campo DUE, prima della operazione		0101 1101	0110 0110

- campo DUE, dopo l'operazione

0110 1101	0110 1001

- Si volevano invertire il 3°, il 4° e gli ultimi 4 bit del campo DUE.

4.12 - EDT : edizione

Per la preparazione di righe di stampa facilmente leggibili, partendo da dati di formato adatto alle operazioni interne, si può utilizzare una apposita istruzione.
Dovendo ad esempio, stampare il risultato di un calcolo, si desidera che non siano stampati gli zeri alla sinistra della prima cifra significativa. Inoltre può essere desiderabile interporre tra le cifre segni di punteggiatura, per esempio la virgola decimale.

CARATTERISTICHE - Il campo 1 contiene la maschera secondo cui deve essere fatta la preparazione del campo 2.
I caratteri da porre nel campo 1 appartengono a queste classi:

a) carattere "SST" codice 0010 0000 (sostituzione),

b) carattere "TSZ" codice 0010 0001 (sostituzione e termine soppressione zeri),

c) carattere "RSZ" codice 0010 0010 (ripristino soppressione zeri),

d) qualsiasi altro carattere.

Questi caratteri producono effetti diversi se vengono trattati in condizione di soppressione zeri o meno.

Il primo carattere del campo 1 è usato inoltre come carattere di riempimento.
L'operazione inizia sempre in condizione di soppressione zeri.
Il campo 2 contiene i dati da preparare alla stampa.

ISTRUZIONE - F = 1101 1110
L1 = lunghezza campo 1
I1 = indirizzo campo 1
I2 = indirizzo campo 2

SVOLGIMENTO - L'operazione procede per indirizzi crescenti: ed inizia in condizione di soppressione zeri.
All'inizio dell'operazione si pongono automaticamente le condizioni RIPO = 1 e DIVE = 0.
In condizione di soppressione zeri i caratteri della maschera posti nel campo 1, provocano i seguenti effetti:

a) SST in 1 provoca la lettura di un carattere del campo 2.
Se i 4 bit meno significativi di tale carattere sono 0000, il carattere di 1 viene sostituito dal carattere di riempimento.
Se viceversa i 4 bit sono diversi da 0000, il carattere di 1 viene sostituito dal carattere di 2 e termina la condizione di soppressione zeri.

b) TSZ in 1 provoca la lettura di un carattere del campo 2 che viene trasferito in 1.
Si ha inoltre il termine della condizione di soppressione zeri.

c) RSZ, o qualsiasi altro carattere, in 1 vengono sostituiti dal carattere di riempimento.

In condizione di non soppressione zeri i caratteri della

maschera di normalizzazione posti nel campo 1 provocano i seguenti effetti:

a) e b) SST e TSZ in 1 provocano la lettura di un carattere del campo 2 che viene trasferito in 1.

c) RSZ ripristina la condizione di soppressione zeri, e viene sostituito dal carattere di riempimento.

d) Qualsiasi altro carattere in 1 permane intatto.

RISULTATO - Il campo 1 contiene i dati nel formato adatto alla stampa. Il campo 2 permane intatto.
Al termine dell'operazione è RIPO = 1. Se DIVE = 0 l'operazione è terminata in condizione di spppressione zeri; se DIVE = 1 l'operazione è terminata in condizione di non soppressione zeri.

I° ESEMPIO :

Esecuzione di una edizione del campo MARIA sul campo ROSA, essendo il campo ROSA di 9 caratteri.

EDT ROSA , MARIA

- campo MARIA

numero				numero		
z0	z7	z0	z8	z0	z4	z0

- campo ROSA prima dell'operazione

%	SST	SST	SST	SST	RSZ	SST	SST	TSZ

- contenuto del campo ROSA dopo l'operazione

ↄ	ↄ	Z7	z0	z8	ↄ	ↄ	z4	z0

stampato come:

		7	0	8			4	0

Con la maschera usata, se il primo dato numerico fosse tutto uguale a zero, lo si trasformerebbe completamente in spazi (ↄ , che non vengono stampati). Il secondo dato numerico invece, essendo stato inserito un carattere TSZ in corrispondenza della sua cifra meno significativa nella maschera, non sarà mai completamente cancellato, anche se completamente uguale a zero.

II° ESEMPIO :

Esecuzione di una edizione del campo LUCIA sul campo ANNA, essendo il campo ANNA di 9 caratteri.

```
EDT     ANNA , LUCIA
```

- campo LUCIA

dato numerico con decimali

a)

z1	z3	z0	z0	z5	z2

b)

z0	z3	z0	z0	z5	z2

c)

z0	z0	z0	z0	z5	z2

↑ cifra migliaia ↑ cifra unità

- campo ANNA

ƀ	SST	.	SST	SST	TSZ	,	SST	SST

- contenuto del campo ANNA dopo l'operazione

a)

ƀ	1	.	z3	z0	z0	,	z5	z2

b)

ƀ	ƀ	ƀ	z3	z0	z0	,	z5	z2

c)

ƀ	ƀ	ƀ	ƀ	ƀ	0	,	z5	z2

stampati rispettivamente come:

a)

	1	.	3	0	0	,	5	2

b)

			3	0	0	,	5	2

c)

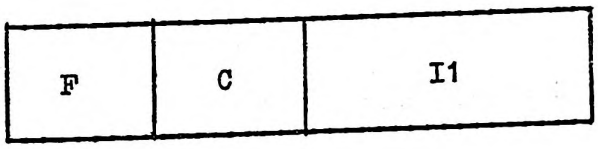
					0	,	5	2

CAP. 5 ISTRUZIONI DI TIPO IMMEDIATO

5.1 - Generalità

Permettono di operare un dato di un carattere contenuto nell'istruzione stessa, con un campo di un carattere situato in memoria.

Il formato dell'istruzione che è a lunghezza fissa di 4 caratteri è il seguente:



- il primo carattere è il codice di funzione F.
  In esso i 2 bit significativi hanno il valore 10 ed i rimanenti 6 bit esprimono il codice vero e proprio.
- il secondo carattere C contiene l'operando costante.
- il terzo ed il quarto carattere indicano l'indirizzo del campo 1.

5.2 - TCF : Trasferimento di un carattere

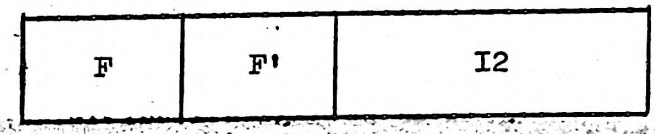
Il contenuto del carattere C dell'istruzione viene posto all'indirizzo I1.
Le condizioni di salto restano immutate.

CAP. 6 ISTRUZIONI DI SALTO

6.1 - Generalità

Le istruzioni di salto permettono di trasferire la esecuzione del programma ad una istruzione qualsiasi interrompendo lo svolgimento sequenziale del programma.

Il formato delle istruzioni di salto, di lunghezza fissa 4 caratteri, è il seguente:



- il primo carattere è il codice di funzione F. In esso i due bit più significativi hanno valore 01 ed i rimanenti 6 bit esprimono il codice vero e proprio;
- il secondo carattere F' è un complemento del codice di funzione;
- il terzo ed il quarto carattere indicano l'indirizzo a cui si trasferisce la esecuzione del programma, nel caso che il salto sia effettuato. Se il salto non è effettuato, l'esecuzione del programma prosegue in modo sequenziale.

6.2 - SAC : salto su condizione

Il salto si effettua se è verificata la condizione indicata nell'istruzione.

CARATTERISTICHE - I 4 bit più significativi del carattere F' indicano la condizione da esaminare: se essa è verificata, esegue il salto di programma. I 4 bit meno significativi devono essere 0000.

La condizione esaminata è una combinazione di configurazione di RIPO e DIVE.

Il salto avviene se:

RIPO = 0 , DIVE = 0	ed il bit 7 di F' è = 1
RIPO = 0 , DIVE = 1	ed il bit 6 di F' è = 1
RIPO = 1 , DIVE = 0	ed il bit 5 di F' è = 1
RIPO = 1 , DIVE = 1	ed il bit 4 di F' è = 1

Per i bit 7,6,5,4 di F' sono possibili in tutto 16 configurazioni, che corrispondono ai seguenti salti:

bit 7	6	5	4	
0	0	0	0	Salto mai verificato
0	0	0	1	Salto verif. se RIPO=1 , DIVE=1
0	0	1	0	Salto verif. se RIPO=1 , DIVE=0
0	0	1	1	Salto verif. se RIPO=1 (DIVE qualsiasi)
0	1	0	0	Salto verif. se RIPO=0 , DIVE=1
0	1	0	1	Salto verif. se DIVE=1 (RIPO qualsiasi)
0	1	1	0	Salto verif. se RIPO ≠ DIVE
0	1	1	1	Salto verif. in tutti i casi fuorchè RIPO=0 , DIVE=0
1	0	0	0	Salto verif. se RIPO=0 , DIVE=0
1	0	0	1	Salto verif. se RIPO = DIVE
1	0	1	0	Salto verif. se DIVE=0 (RIPO qualsiasi)
1	0	1	1	Salto verif. in tutti i casi fuorchè RIPO=0 , DIVE=1
1	1	0	0	Salto verif. se RIPO=0 (DIVE qualsiasi)
1	1	0	1	Salto verif. in tutti i casi fuorchè RIPO=1 , DIVE=0
1	1	1	0	Salto verif. in tutti i casi fuorchè RIPO=1 , DIVE=1
1	1	1	1	Salto verif. sempre.

ISTRUZIONE - F· = 0100 0011
F' = vedere precedente tabella
I2 = indirizzo della istruzione alla quale si salta

SVOLGIMENTO - Se la condizione indicata in F' è verificata il salto si svolge, altrimenti il programma prosegue esaminando l'istruzione successiva a quella di salto.

RISULTATO - L'istruzione di salto non modifica il contenuto di RIPO e DIVE.

6.3 - SAR : salto e riserva indirizzo

Il salto invia sistematicamente la esecuzione del programma all'indirizzo indicato nell'istruzione.

CARATTERISTICHE - F' ha valore 1111 0000. L'istruzione produce l'immagazzinamento dell'indirizzo di programma della istruzione che segue la SAR in memoria. L'indirizzo viene depositato nelle posizioni 254 e 255 della memoria.

ISTRUZIONE - F· = 0100 0001
F' = 1111 0000
I2 = indirizzo della istruzione alla quale si salta.

SVOLGIMENTO - Il salto avviene sempre.

RISULTATO - L'istruzione non modifica il contenuto di RIPO

e DIVE, e lascia l'indirizzo della istruzione seguente nelle posizioni 254 e 255 della memoria.

6.4 - SAV: salti su condizioni di console od errore

Il salto avviene se sono verificate le condizioni indicate nell'istruzione.

CARATTERISTICHE - Il carattere F' che specifica la condizione può assumere solo i seguenti valori:

- F' - 1000 0000 : salto se è abbassata la chiave COES 1 sulla console,
- F' - 0100 0000 : salto se è abbassata la chiave COES 2 sulla console,
- F' - 0010 0000 : salto se è disposta la condizione di errore interno, e cancellazione della condizione di errore interno.

ISTRUZIONE - F = 0101 0011
F' = vedere precedenti valori
I2 = indirizzo della istruzione alla quale si salta.

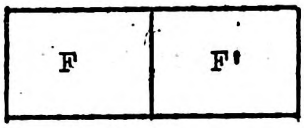
SVOLGIMENTO - Se le condizioni indicate in F' sono verificate il salto si svolge, altrimenti il programma prosegue esaminando l'istruzione successiva a quella di salto.

RISULTATO - L'istruzione non modifica il contenuto di RIPO e DIVE, e nel 3° caso indicato cancella la condizione di errore interno.

CAP. 7 ISTRUZIONI VARIE

7.1 - Generalità

Il formato dell'istruzione, che è a lunghezza fissa di due caratteri, è il seguente:



- il primo carattere è il codice di funzione F. In esso i 2 bit più significativi hanno il valore 00, ed i rimanenti 6 bit esprimino il codice vero e proprio;
- il secondo carattere F' è il complemento del codice di funzione ed ha sempre valore 0000 0000.

7.2 - ALT : istruzione di fermata

L'istruzione ALT produce, al termine della lettura del carattere F, l'arresto dell'elaborazione; essa riparte premendo il tasto del VIA sul quadro di comando.

F = 0000 1010

F' = 0000 0000

Le condizioni RIPO e DIVE permangono inalterate.

7.3 - NOP : nessuna operazione

L'istruzione non esegue alcuna operazione ed il programma procede ad esaminare l'istruzione successiva.

F = 0000 0111

F' = 0000 0000

Le condizioni RIPO e DIVE permangono inalterate.

CAP. 8 ISTRUZIONE ESTERNA

8.1 - Collegamento delle unità esterne

Prima di descrivere la istruzione esterna, la quale agisce sulle unità periferiche collegate, si richiamano le possibilità di collegamento fra queste ultime la cui trattazione è stata fatta nel capitolo "unità centrale".

La unità centrale 4.115 presenta 4 connettori di attacco per unità periferiche:

- connettore specializzato per il collegamento di una stampante parallela tramite un governo integrato, oppure per il collegamento di una stampante seriale, tramite governo di tipo standard;
- connettore specializzato per il collegamento di un lettore seriale di schede perforate o di documenti magnetici, tramite un governo integrato;
- connettore di tipo standard per il collegamento diretto di una unità periferica qualsiasi con governo di tipo standard, oppure di un numero superiore (fino a 64) di unità periferiche di tipo standard, tramite sincronizzatori 4000;
- connettore di tipo standard (opzionale) per il collegamento diretto di una unità periferica qualsiasi con governo standard.

I governi delle unità connesse ai primi due connettori sono detti integrati, in quanto parte delle funzioni di governo di tali unità sono rimandate alla unità centrale.

Le operazioni periferiche non sono in generale sovrapponibili, durante le fasi di trasferimento dati, con il calcolo.

Sono però sovrapponibili tra loro, salvo vincoli di frequenza massima di trasferimento, due operazioni periferiche qualsiasi tra le seguenti:

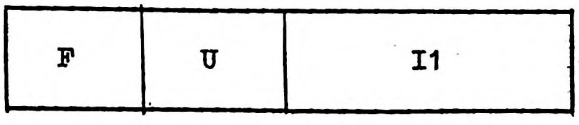
- stampa della stampante parallela integrata,
- lettura dal lettore integrato,
- trasferimento di dati da o verso una delle unità non integrate.

I canali 1 e 2, attraverso i quali avvengono gli scambi delle informazioni, sono dotati, ciascuno, di un indicatore di errore di trasferimento.

L'indicatore è posizionato nel caso di trasmissione di un carattere di parità errata, e deve essere esaminato dopo ogni trasmissione di dati in ingresso o uscita, assieme alle altre indicazioni di condizione anormale associate alle singole unità esterne.

8.2 - EST : istruzione esterna

Questa classe è costituita da una sola istruzione, la quale presenta il seguente formato di 4 caratteri:



- il primo carattere è il codice di funzione F. In esso i due bit più significativi hanno il valore 10,
- il secondo carattere U definisce l'unità esterna interessata dalla istruzione,

- il terzo ed il quarto carattere indicano l'indirizzo I1 del campo 1 il quale specifica l'operazione che l'unità esterna deve svolgere.

L'operazione esterna indicata nel campo 1 può essere di tre tipi:

- comando a unità esterna,
- esame di condizione esterna,
- trasferimento a o da unità esterna.

8.2.1 - Comando a unità esterna

Ordina all'unità esterna l'esecuzione di una funzione che non implica il trasferimento di dati.

CARATTERISTICHE - Il campo 1 ha il formato ZX, in cui il carattere Z specifica il tipo di operazione ed alcune modalità con cui l'operazione deve essere eseguita ed il carattere X è un codice di comando che viene inviato all'unità esterna, e da essa è interpretato ed eseguito (esempio : riavvolgimento nastro, selezione di caselle di ricezione, ecc.)
Le diverse configurazioni che il carattere Z può assumere sono indicate nella tabella che segue. Esso permette di definire:

- se la operazione da eseguire è l'invio di un comando a unità esterna (v. bit 7,6);
- se l'invio del comando deve avvenire dopo che il governo della unità periferica abbia terminato l'eventuale operazione in corso, dichiarandosi quindi disponibile (v. bit 2);

- il canale (1 oppure 2) su cui viene eseguita la operazione. Per questo tipo di operazione la indicazione del canale serve solo a determinare quale delle due condizioni di errore trasferimento, associate una al canale 1 e l'altra al canale 2, deve essere annullata (v. bit 0).

Il carattere X indica il tipo di comando (esempio: riavvolgimento nastro, selezione di caselle di ricezione, ecc.).

Il valore del codice X dipende dal tipo di unità periferica a cui viene inviato l'ordine.

Bit del carattere Z (comando ad unità esterna)

7	6	5	4	3	2	1	0
1	0	0	0	0	0/1 0 = attesa unità libera 1 = esecuzione immediata	0	0/1 0 = canale 1 1 = canale 2

SVOLGIMENTO - L'istruzione viene interpretata e viene letto il campo 1.
RIPO e DIVE vengono posti automaticamente ad uno e a zero.
Viene annullata la condizione di errore di trasferimento associata al canale interessato.
Il comando, corrispondente al carattere X, viene inviato alla unità esterna e la istruzione termina non appena il carattere è stato accettato.
Normalmente la accettazione di un comando richiede un tempo assai piccolo, mentre la vera e propria esecuzione può poi avvenire in seguito, contemporaneamente a quella delle istruzioni successive del programma (esempio: salto di carta o interlinea della stampante).

8,2.2 - Esame di condizione esterna

Esamina se è verificata sulla unità esterna, una delle condizioni che interessano il programma (esempio: unità occupata, errore, casella di ricezione schede piena, fine nastro, ecc.).

CARATTERISTICHE - Il campo 1 ha il formato ZX, in cui il carattere Z specifica il tipo di operazione ed alcune modalità di esecuzione; il carattere X specifica quale delle diverse condizioni, che definiscono lo stato della unità esterna, deve essere esaminata.
Le diverse configurazioni che il carattere Z può assumere sono indicate nella tabella che segue. Esso permette di definire:

- che la operazione da eseguire è un esame di condizione esterna (v. bit 7,6);
- se si vuole o meno che l'esame della condizione esterna sia preceduta dalla attesa che il governo della unità periferica si dichiari disponibile (v. bit 2). Questa attesa potrà essere utilizzata per far avvenire corretta

mente l'esame di alcune condizioni (per esempio di errore, fine supporto, ecc.) che devono essere esaminate solo al termine di una operazione esterna indipendente (esempio: una operazione di unità con buffer), durante la quale il governo si dichiara non disponibile;

- il canale (1 oppure 2) su cui viene eseguita la operazione. Per questo tipo di operazione la indicazione del canale serve solo a determinare quale delle due condizioni di errore trasferimento, associate ai canali 1 e 2, dovrà essere sentita (qualora X specifichi l'esame delle condizioni di errore).

Il valore del codice X dipende dal tipo di unità periferica interessata, e dalla condizione che si vuole esaminare.

Bit del carattere Z (esame di condizione esterna)

7	6	5	4	3	2	1	0
1	1	0	0	0	0/1 0 = attesa unità libera 1 = esecuzione immediata	0	0/1 0 = canale 1 1 = canale 2

SVOLGIMENTO - La istruzione viene interpretata ed è letto il campo 1.
RIPO e DIVE vengono posti automaticamente ad uno e a zero.
Viene esaminata la condizione indicata dal carattere X. Se tale condizione è presente, viene posto DIVE = 1, altrimenti risulta DIVE = 0.
DIVE potrà poi essere esplorato tramite un salto su condizione.

8.2.3 - Trasferimento dati

Comanda il trasferimento dei dati tra la memoria centrale e le unità esterne. Questi trasferimenti avvengono attraverso due canali indipendenti, detti canale 1 e canale 2, che possono operare uno per volta o in sovrapposizione.

CARATTERISTICHE - Il campo 1 ha il formato ZXLI (6 caratteri), in cui il carattere Z specifica il tipo di operazione e le modalità di esecuzione.
Il carattere X contiene un codice di comando che viene inviato alla unità periferica all'inizio della operazione, ed è da essa interpretato come comando di avvio della operazione.
Il campo L, di 2 caratteri, indica ~~con i 12 bit meno significativi~~, una lunghezza massima di trasferimento.
~~I 4 bit più significativi del primo carattere L devono valere 0.~~ La lunghezza deve essere indicata come la differenza fra gli indirizzi esterni dell'area di memoria utilizzata per i dati, cioè come lunghezza in caratteri di tale area diminuita di 1.
Sono perciò indicabili le lunghezze da 1 (L = 0000 0000 0000 0000) a 4096 (L = 0000 1111 1111 1111) caratteri.
Il campo I di 2 caratteri indica ~~l'indirizzo del primo carattere in memoria interessato dal trasferimento (secondo~~ il formato usuale degli indirizzi).

Vedi correzione allegata a pag. 83

Con il carattere Z, le cui configurazioni sono indicate nella tabella che segue, si può specificare:

- che la operazione da eseguire è un trasferimento di dati (v. bit 7);
- se il trasferimento è in ingresso o in uscita da calcolatore (v. bit 6);
- se durante il trasferimento la evoluzione dell'indirizzamento deve avvenire nel verso degli indirizzi crescenti o decrescenti, (v. bit 5);
- se i caratteri del supporto esterno corrispondono a byte o semibyte di memoria (v. bit 4);
- se la istruzione deve solo predisporre la operazione oppure è esecutiva (nel senso che sarà tra poco descritto) (v. bit 3);
- se si deve attendere la disponibilità della unità esterna prima di eseguire la operazione (v. bit 2);
- se il trasferimento è richiesto sul canale 1 oppure 2. (v. bit 0)

7	6	5	4	3	2	1	0
0	0/1 0 - ingresso 1 - uscita	0/2 0 - per indirizzi crescenti 1 - per indirizzi decrescenti	0/1 0 - 4 bit + 4 bit per byte 1 - 8 bit per byte	0/1 0 - istruzione esecutiva 1 - istruzione di predispos.	0/1 0 - attesa di unità libera 1 - esecuzione immediata	0	0/1 0 - canale 1 1 - canale 2

E' possibile eseguire dei trasferimenti di dati da memoria a unità periferica o inversamente, utilizzando dei byte di memoria consecutivi nel verso degli indirizzi crescenti o decrescenti. Sul canale 2 è però possibile operare solo nel verso degli indirizzi crescenti.

Per ragioni di capacità di controllo associate ai due canali 1 e 2, sul canale 2 possono essere eseguite solo le operazioni di stampa della stampante parallela integrata o di lettura dal lettore integrato. Sul canale 1 possono essere eseguite le operazioni di lettura dal lettore seriale integrato ~~o di stampa della stampante seriale integrata~~, o di trasferimento in ingresso da, o in uscita verso, una delle unità con governo standard. (eccetera la stampante seriale collegabile tramite governo standard al canale 1)

Mediante una istruzione di tipo esecutivo (v.bit 3) è possibile avviare immediatamente una operazione di trasferimento di dati su uno dei due canali, durante la cui esecuzione non avvengono altre operazioni.

Programmando invece, consecutivamente e nell'ordine, una istruzione di predisposizione sul canale 2 ed una istruzione esecutiva sul canale 1 si ottiene l'avviamento contemporaneo delle due operazioni richieste; al termine della più lunga delle due riparte la esecuzione del programma.

Non ha senso una istruzione di predisposizione sul canale 1.

Per facilitare il caricamento dei programmi è previsto un modo di operazione esterna in cui ad ogni byte di memoria vengono fatti corrispondere due caratteri di supporto esterno, aventi solo 4 bit significativi ciascuno.

Questo modo di operazione è possibile solo per l'ingresso di dati sul canale 1 ed una operazione di tale tipo ("ingresso con impaccamento") non può essere sovrapposta ad una operazione sul canale 2.

Nell'ingresso con impaccamento ogni coppia di caratteri

consecutivi provenienti dall'esterno viene utilizzata così: i 4 bit meno significativi del primo e del secondo carattere sono utilizzati per formare i 4 bit più ed i 4 bit meno significativi di un byte, che viene scritto in memoria. Se l'ingresso termina dopo un numero dispari di caratteri, l'ultimo dei caratteri viene scritto per intero in un ultimo byte di memoria, come avverrebbe in un ingresso normale. La lunghezza indicata nel campo 1 si riferisce ~~in questo caso al numero di caratteri sul supporto esterno~~. al numero di byte di memoria impiegati.

Le operazioni sul canale 2 sono sottoposte a vincoli di indirizzamento; definita "riga" di memoria un gruppo di 256 byte consecutivi, con indirizzo di sinistra 0, o 256, o multiplo di 256, le limitazioni possono essere così espresse:

- Per la stampa da parte della stampante parallela integrata, la zona di memoria interessata deve iniziare due caratteri a destra dell'estremo sinistro di una riga, I due caratteri lasciati a sinistra vengono utilizzati per governare la stampa, e non devono pertanto essere utilizzati come deposito di dati durante la operazione, il loro contenuto al termine di una stampa è indefinito.
  La limitazione può essere espressa così: ogni ordine di stampa deve essere dato con un indirizzo del tipo:

  $K \cdot 256 + 2$ con $K = 0, 1, 2, \ldots,$

  Dopo la stampa i 2 caratteri agli indirizzi $K \cdot 256$ e $K \cdot 256+1$ (con il valore di K scelto) hanno un contenuto indefinito, mentre la riga di stampa permane intatta in memoria.
  La stampa avviene sempre per una lunghezza pari al numero di colonne della stampante usata: tale lunghezza, diminuita di 1, deve essere espressa nel campo 1 della istruzione (carattere L).
  Per ottenere la stampa di una riga parziale, le zone di memoria corrispondenti alle colonne di stampa su cui non

si desidera stampare vanno riempite di caratteri "spazio" (~~nel codice accettato dalla stampante~~). (5)

- Per la lettura da parte del lettore seriale integrato, sul canale 2, le modalità sono diverse a seconda del tipo di lettore collegato.

A) Lettore di schede.

La lettura avviene nell'ordine da colonna 1 a 2,..... .. a 80, cioè da sinistra a destra, e l'ingresso avviene nel verso degli indirizzi crescenti.
L'ordine di lettura deve essere dato con un indirizzo del tipo:

$$K \cdot 256 + 2 \qquad \text{con } K = 0, 1, 2, \ldots$$

La lunghezza del trasferimento è determinato dal modo di lettura: 80 caratteri per la lettura di schede normali, 160 per la lettura di schede binarie (che originano 2 caratteri per ogni colonna di scheda; la lettura binaria è consentita da un dispositivo opzionale su alcuni lettori).
Nei due casi devono essere specificate, nei caratteri di L, le lunghezze 79 (= 80-1) e 159 (=160-1) rispettivamente.

B) Lettore di documenti con caratteri magnetici.

La lettura viene fatta da destra a sinistra e l'ingresso avviene nel verso degli indirizzi decrescenti.
Detta LD la lunghezza massima in caratteri dei documenti da leggere, inclusi tutti i caratteri di servizio e l'eventuale carattere di fine riga generato dal lettore, l'ordine di lettura deve essere dato con un indirizzo del tipo:

$$K \cdot 256 + LD + 1 \qquad K = 0, 1, 2, \ldots$$

e con una lunghezza (caratteri di L):

$$LD - 1$$

La introduzione del contenuto avviene allora fra l'indirizzo precisato nella istruzione e l'indirizzo:

K · 256 + 2 (ultimo carattere entrato)

Esempio: nel caso di un documento marcato su una sola riga, che genera in lettura al massimo 64 caratteri (LD = 65), deve essere:

$$I = K \cdot 256 + 66 \quad , \quad L = 64$$

Le operazioni sul canale 1 non sono sottoposte a vincoli di indirizzo, ed avvengono per una lunghezza che è la minore tra le due quantità seguenti:

- contenuto del campo 1 (caratteri L) aumentato di 1,
- numero di caratteri trasferito fino all'istante in cui dalla unità esterna operante perviene una segnalazione di fine trasmissione (es.: fine blocco di nastro in lettura, fine scheda).

SVOLGIMENTO - Istruzione di predisposizione sul canale 2.
La istruzione viene interpretata ed è letto il campo 1.
RIPO e DIVE vengono posti automaticamente ad uno e a zero.
Viene esaminata (ciò deve sempre essere richiesto nel carattere Z) la disponibilità della unità, attendendo che la unità si renda disponibile se non lo è già immediatamente. Viene annullata la condizione di errore trasferimento sul canale 2.
Il comando di inizio operazione viene memorizzato e sarà reso esecutivo dalla istruzione seguente in programma (che deve essere una istruzione esterna di trasmissione sul canale 1).
Si passa quindi alla interpretazione della istruzione seguente.

- Istruzione esecutiva sul canale 2.

La istruzione viene interpretata ed è letto il campo 1.
RIPO e DIVE vengono posti automaticamente ad uno e a zero.
Viene esaminata (ciò deve essere sempre richiesto nel carattere Z) la disponibilità della unità, attendendo che la

unità si renda disponibile se non lo è già immediatamente.
Viene annullata la condizione di errore trasferimento sul canale 2.
Viene inviato alla unità esterna interessata il comando di inizio operazione, ed ha luogo la operazione (stampa o lettura).
Al termine della operazione si passa alla istruzione seguente.

- Istruzione sul canale 1, non preceduta da predisposizione del canale 2.

La istruzione viene interpretata ed è letto il campo 1.
RIPO e DIVE vengono posti automaticamente ad uno ed a zero.
Viene esaminata, se ciò è richiesto nel carattere Z, la disponibilità della unità, attendendo che la unità si renda disponibile se non lo è già immediatamente.
Viene annullata la condizione di errore trasferimento sul canale 1.
Viene inviato all'unità il comando di inizio operazione e ha luogo l'operazione.
Terminata l'operazione, viene scaricato agli indirizzi 254 e 255 l'indirizzo del carattere seguente l'ultimo interessato dal trasferimento, nel verso in cui è avvenuto il trasferimento.
Se si è ricevuta dalla unità esterna una segnalazione di fine si pone DIVE = 1, altrimenti resta DIVE = 0.
Si passa quindi alla istruzione seguente.

- Istruzione sul canale 1 preceduta da predisposizione sul canale 2.

La istruzione viene interpretata ed è letto il campo 1.
RIPO e DIVE vengono posti automaticamente ad uno e a zero.
Viene esaminata, se ciò è richiesto, la disponibilità della unità, attendendo che la unità si renda disponibile se non lo è già immediatamente.
Viene annullata la condizione di errore trasferimento sul canale 1.
Vengono inviati i comandi di inizio delle due operazioni richieste sui canali 1 e 2; le due operazioni sono eseguite contemporaneamente.
Terminata la più lunga delle due operazioni, viene scarica

to agli indirizzi 254 e 255 l'indirizzo del carattere seguente l'ultimo interessato dal trasferimento sul canale 1, nel verso in cui è avvenuto il trasferimento.
Se si è ricevuta dalla unità esterna, operante sul canale 1, una segnalazione di fine, si pone DIVE = 1, altrimenti resta DIVE = 0.
Si passa quindi alla istruzione seguente.

Valori del codice U

Valori da 0000 0000 a 0011 1111	selezione di una delle 64 possibili unità collegate al 1° connettore standard
0100 0000	selezione della unità collegata al 2° connettore standard
1000 0000	selezione del lettore integrato
1100 0000	selezione della stampante integrata

VALORI DEL CODICE X PER I COMANDI A UNITA' ESTERNA

0000 1010	Interlinea
0000 1100	Alimenta supporto
0001 0001	Avvio unità meccanica
0001 0100	Arresto unità meccanica
0100 0000	Leggi avanti
0100 0010	Registra avanti, o stampa
0100 0011	Stampa con caratteri magnetici
0100 0111	Reset errore generale
0100 1000	Espelli supporto
0101 0001	Salto verticale 1
0101 0010	Salto verticale 2
0101 0111	Salto verticale 7
0101 1001	Interlinea doppia
1010 0000	Set modo
1010 0001	Reset modo

VALORI DEL CODICE X PER LA SELEZIONE DELLE CONDIZIONI ESTERNE

Binario	nome standard	Significato		
		Standard	Lettore	Stampante
0000 0001	PE00	Governo pronto	Pronto alla lettura	Stampante libera
0000 0011	SEGE	1° errore	Errore	-
0000 0101	FISE	Fine sequenza	FINE SEQUENZA -	-
0000 1011	EGOL	2° errore	-	-
0001 0000	MAPE	Unità pronta	Pronto alla aliment.	-
0001 0010	TESE	Fine supporto	Fine schede	Fine moduli
0001 0100	MARE	Fuori servizio	Fuori servizio	Fuori servizio
0001 1010	MATE	Macchina attivata	-	-
0001 1100	INES	-	-	-
0001 1110	CAPE	Ulteriore condiz.anormale	Casella piena	-
0010 0010	IGOL		-	-
0010 0100	NUIO	N° unità (1° bit)	-	-
0010 1010	NU20	N° unità (2° bit)	-	-
0010 1100	NU30	N° unità (3° bit)	-	-
0010 1110	SECO	Funzionamento in manuale	Funzionam. man.	Funzionam. man.
0100 0000	ZI03	-	-	-
0100 0010	-	Errore trasferimento (ERER o ERAR)		
0100 0100	-	Or delle condizioni anormali (SEGE, FISE, EGOL, TESE, MARE, CAPE, ERAR, ERER)		
0100 1010	-	Or degli errori (SEGE, EGOL, ERER, ERAR)		

CAP. 9 LA CONSOLE OPERATIVA

Per la messa a punto dei programmi e per le normali funzioni operative esiste una console operativa la quale presenta i seguenti dispositivi di comando:

- TENS — Chiave con segnalazione luminosa incorporata, per dare tensione alla macchina. La luce indica "macchina sotto tensione".

- REGE — Pulsante di sgancio generale all'unità centrale o ai governi esterni. Va azionato ogni volta che venga data tensione alla macchina ed ogni volta che, da una situazione indefinita, si desidera riportare la macchina in condizioni di riprendere un funzionamento definito.
  La pressione dello sgancio durante il funzionamento può distruggere parzialmente il contenuto della macchina. Dopo tale manovra la unica operazione sensata è il ricaricamento del programma.
  La pressione dello sgancio a macchina ferma (per un ALT interno o per l'azionamento dello STOP) ha invece il solo effetto di cancellare le segnalazioni di errore e riportare all'indirizzo zero la esecuzione del programma, che può correttamente riprendere con una pressione del VIA.

- VIA — Pulsante che fa partire la esecuzione del programma dopo una fermata per istruzione ALT o per pressione dello STOP.

- INPI — Pulsante che predispone il caricamento di una scheda di programma dal lettore integrato. Il successivo azionamento del pulsante "VIA" fa iniziare la sequenza di ingresso.

- INPE — Pulsante che predispone il caricamento di un

blocco di programma dalla unità di nome 0000 0000 collegata al terzo connettore.
Il successivo azionamento del pulsante "VIA" fa iniziare la sequenza di ingresso.

- STOP Chiave con segnalazione luminosa incorporata che provoca uno stop della esecuzione del programma e dispone il calcolatore in condizione di arrestarsi a ogni istruzione dopo la lettura del codice di funzione.
La fermata per effetto dello STOP avviene sempre dopo il primo ciclo di interpretazione di una istruzione (lettura del carattere F).
Sulla console sono visualizzati l'indirizzo del carattere F letto (V0) ed il suo valore (F0).
Con la chiave STOP abbassata, premendo il VIA si ottiene la esecuzione di una sola istruzione, e ci si ferma all'inizio della interpretazione della successiva.
La luce incorporata si accende quando la chiave è abbassata.

- EA Lampadino indicante errore di parità nel trasferimento sul canale 2.

- EE Lampadino indicante errore di parità nel trasferimento sul canale 1.

- EI Lampadino indicante errore di parità in un dato letto da memoria. Questo errore provoca una fermata immediata della elaborazione.

- ALT Segnalazione di temporizzazione ferma. Si accende dopo una fermata della elaborazione. Per far ripartire la temporizzazione è necessario agire sul tasto VIA, a meno che la fermata sia avvenuta per errore interno.

- ATE Segnalazione di temporizzazione ferma per atte

sa di ingresso da unità esterna.
La fermata della macchina con questa luce accesa indica una condizione anormale su qualche unità esterna.
Ulteriori segnalazioni luminose sono contenute nei tasti REGE e VIA.
Nel tasto REGE è incorporata la segnalazione di macchina in manutenzione, che garantisce l'esatto posizionamento delle chiavi di console di manutenzione per il funzionamento normale.

INIZIALIZZAZIONE - L'inizio delle operazioni della macchina va ottenuto con le seguenti manovre:

- sgancio
- pressione di uno dei due pulsanti di inizializzazione
- via

La operazione di inizializzazione consiste nella lettura di un blocco di informazione, da utilizzarsi come programma di avvio, da un supporto esterno, e nella partenza dell'esecuzione del programma dall'inizio del programma introdotto.

La lettura viene eseguita:

- dal lettore integrato, se si preme il primo dei pulsanti di inizializzazione;
- dalla unità di codice 0000 0000, collegata al 1° bocchettone standard, se si preme il secondo.

La lettura avviene a partire dall'indirizzo zero di memoria, nel modo "ingresso per indirizzi crescenti con impacamento", per una lunghezza pari a un blocco di informazioni sul supporto esterno (scheda o blocco di nastro o documento CMC7) ed in ogni caso non superiore a 1024 caratteri di supporto (512 byte).
La esecuzione del programma parte poi, al termine dell'ingresso dall'indirizzo zero.

CAP. 10 NORME DA SEGUIRE NELLA COMPILAZIONE DEI PROGRAMMI

Le regole da seguirsi nella programmazione, tenendo presente le diverse capacità che la memoria della macchina può presentare, sono le seguenti:

- i campi indirizzo di istruzione e di pseudoistruzione esterna dovranno avere sempre la forma:

  0000 XXXX XXXX XXXX ,

  per la versione di macchina a 4096 (o meno) posizioni di memoria e

  000X XXXX XXXX XXXX ,

  per la versione di macchina a 8192 posizioni di memoria.

- Nelle versioni macchina fino a 4096 posizioni di memoria l'indirizzamento della memoria è circolare, passando dalla posizione 4095 alla 0 e viceversa. Nella versione a 8192 posizioni l'indirizzamento è circolare dalla posizione 8191 alla 0 e viceversa.

- Nella programmazione delle operazioni di modifica di indirizzo occorrerà fare attenzione a non generare bicaratteri di indirizzo proibiti, partendo da indirizzi permessi.
  Si voglia modificare l'indirizzo 8190 sommandogli 2 per generare, dopo aver attraversato l'indirizzo massimo della memoria, l'indirizzo 0 della stessa.
  La operazione non va eseguita come:

```
0001   1111   1111   1110   +
0000   0000   0000   0010
-------------------------
0010   0000   0000   0000
```

ma come:

```
0001  1111  1111  1110   +
1110  0000  0000  0010
---------------------------
0000  0000  0000  0000
```

Nel secondo modo gli 1^i più significativi provocano la la propagazione del riporto, che viene poi perduto dalla parte più significativa del risultato.

Alla circostanza dell'esempio si ritorna ogni volta, che una modifica di indirizzo porta il valore dell'indirizzo ad attraversare gli estremi della memoria. In tali casi si genera infatti un riporto o un prestito verso le posizioni più significative del bicarattere di indirizzo, che cambiano il loro valore da 000 (o 0000) a 001 (0001) oppure 111 (1111), a meno che non si ritocchi opportunamente (v. sopra) la quantità modificatrice.

- Il campo di lunghezza della pseudoistruzione esterna deve avere sempre la forma:

```
0000  00XX  XXXX  XXXX  ,
```

per la versione di macchina a 4096 (o meno) posizioni di memoria e:

```
0000  XXXX  XXXX  XXXX
```

per la versione di macchina a 8192 posizioni.

VALORI DEI CODICI DI FUNZIONE F ESPRESSI IN BINARIO

	7	6	5	4	3	2	1	0
NOP	0	0	0	0	0	1	1	1
ALT	0	0	0	0	1	0	1	0
SAR	0	1	0	0	0	0	0	1
SAC	0	1	0	0	0	0	1	1
SAV	0	1	0	1	0	0	1	1
TFC	1	0	0	1	0	0	1	0
CFC	1	0	0	1	0	1	0	1
EST	1	0	0	1	1	1	1	0
TFR	1	1	0	1	0	0	1	0
AND	1	1	0	1	0	1	0	0
CFR	1	1	0	1	0	1	0	1
ORL	1	1	0	1	0	1	1	0
ORE	1	1	0	1	0	1	1	1
DSP	1	1	0	1	1	0	0	0
RCA	1	1	0	1	1	0	0	1
IMP	1	1	0	1	1	0	1	0
RCI	1	1	0	1	1	0	1	1
TCD	1	1	0	1	1	1	0	0
EDT	1	1	0	1	1	1	1	0
TFN	1	1	1	1	1	0	0	0
CFN	1	1	1	1	1	0	0	1
ADD	1	1	1	1	1	0	1	0
SOD	1	1	1	1	1	0	1	1
ADB	1	1	1	1	1	1	1	0
SOB	1	1	1	1	1	1	1	1

SIGNIFICATO DI RIPO E DIVE DOPO LE VARIE OPERAZIONI

		00	01	10	11
ADD,ADB (±)	$I_1 + I_2 \rightarrow I_1$	$I_1 = 0$	$I_1 \neq 0$	$I_1 = 0$ OVER FLOW	$I_1 \neq 0$ OVER FLOW
SOD,SOB	$I_1 - I_2 \rightarrow I_1$		$I_1 < 0$	$I_1 = 0$	$I_1 > 0$
CFN,CFR,CFC (±)	$I_1 :: I_2$ o C		$I_1 < I_2$ o C	$I_1 = I_2$ o C	$I_1 > I_2$ o C
TFN	$I_2 \rightarrow I_1$	$I_1 = 0$	$I_1 \neq 0$		
ORE	$I_1 / I_2 \rightarrow I_1$			$I_1 = 0$	$I_1 \neq 0$
RCI,RCA	RICERCA I_2 in I_1			NON TROVATO	TROVATO
EDT	TERMINA CON SOPPRESS. ZERI			SI	NO
EST	ESAME CONDIZ.ESTERNA			NON VERA	VERA
EST	TRASMISSIONE			FINE SU LUNGHEZZA	FINE DA SUPPORTO
EST	COMANDO			SEMPRE	

TUTTE LE ALTRE OPERAZIONI NON TOCCANO RIPO E DIVE

(±) = Il significato di RIPO e DIVE è riferito ai casi di somma, sottrazione, confronto di operandi considerati positivi. Il risultato negativo di sottrazione è lasciato in I_1 in forma complementata.

TEMPI DI ESECUZIONE DELLE ISTRUZIONI IN CICLI DI MEMORIA

Nelle condizioni normali il ciclo di memoria è di 8 usec, ed il coefficiente R è uguale a 0,5.

Istruzioni	n. cicli di memoria	Osservazioni
ADD, SOD ADB, SOB TFN, CFN	6+L2+(1+R).L1	L2 = lunghezza in caratteri di 2 L1 = lunghezza in caratteri di 1
EDT	6+2 ·L1 + N	N = n. di caratteri SST e TSZ in 1 (maschera)
TCD, IMP	6+3· L1	
DSP	6+(3+2R) . L2	
CFR	6+(2+R) . D	D = lunghezza di 2 e 1 da esplorare per trovare diversità, se esiste; altrimenti lunghezza di 2 ed 1.
TFR	6+2 . L	
ORL, ORE AND	6+(2+R) . L	L = lunghezza in caratteri di 2 e 1
RCA, RCI	9 + U	U = lunghezza di 1 da esplorare per trovare eguaglianza, se esiste; altrimenti lunghezza di 1
CFC	5 + R	
TFC	5	

Istruzioni	n. cicli di memoria	Osservazioni
SAC, SAV	4	
SAR	6	
NOP	2	
ALT	2 + TF	TF = tempo di macchina ferma
EST (esame condiz.)	7 + (TU)	TU = tempo di attesa di unità libera (se tale attesa è richiesta)
EST (comando)	7 + (TU) + TC	
EST (trasmissione canale 1)	9 + (TU) + TC + TF	TC = tempo di accettazione del comando, variabile da governo a governo
EST (trasmissione canale 2)	11+(TU) + TC + TF	TF = tempo di esecuzione del trasferimento esterno, variabile da unità a unità. Nel caso di contemporaneità è il più lungo fra i tempi relativi alle due unità impegnate.
EST (predisposizione canale 1)	9 +(TU) + TC	

TABELLE DI CONFIGURAZIONE DEI CARATTERI

Nelle seguenti tabelle sono riportati i codici di scheda Hollerith e Bull.

Codice Hollerith (IBM)

	1	2	3	4	5	6	7	8	9	8 e 2	8e3	8e4	8e5	8e6	8e7	o	/
12	A	B	C	D	E	F	G	H	I		•	]	(	<	\	+	&
11	J	K	L	M	N	O	P	Q	R		$	*	)	;	'	↑	-
o	/	S	T	U	V	W	X	Y	Z	←	,	%	=	"	!		0
/	1	2	3	4	5	6	7	8	9	[	#	@	:	>	?		b

Codice Bull

	11	0	1	2	3	4	5	6	12	/
7	A	B	C	D	E	F	G	H	0	7
8	J	K	L	M	N	P	Q	R	I	8
9	S	T	U	V	W	X	Y	Z	'	9
/	•	0	1	2	3	4	5	6	*	b
9e8	,	(	)	-	+	]	?	@	>	=
9e7	\	#	:	←	"	%	&	;	$	/
o	↑								!	
1	[								<	

Le unità a schede di lettura e di perforazione possono essere collegate all'Elea 4-115 in due modi :

- attraverso un trascodificatore in grado di trascodificare il codice di scheda IBM in codice macchina "GEDC", direttamente operabile.

Codice GEDC

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000					0	b					↑	+				
0001					1	A					J	/				
0010					2	B					K	S				
0011					3	C					L	T				
0100					4	D					M	U				
0101					5	E					N	V				
0110					6	F					O	W				
0111					7	G					P	X				
1000					8	H					Q	Y				
1001					9	I					R	Z				
1010					[	&					-	←				
1011					#	.					$	,				
1100					@	]					*	%				
1101					:	(					)	=				
1110					>	<					;	"				
1111					?	\					'	!				

Il trascodificatore può inoltre essere dotato del dispositivo di by-pass.

- attraverso un sistema condensatore, in grado di trascodificare il codice di scheda Bull in un codice intermedio compatto, convertibile successivamente in codice GEDC.

Codice intermedio compatto

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	11	9/11	7	7/12												
0001	0	9/0	11/0	12/0												
0010	1	9/1	11/1	12/1												
0011	2	9/2	8	8/12												
0100	3	9/3	9/7	9/7/12												
0101	4	9/4	9	9/12												
0110	5	9/5	/	12												
0111	6	9/6	9/8	9/8/12												
1000	7/11	9/7/11	8/11	9/8/11												
1001	7/0	9/7/0	8/0	9/8/0												
1010	7/1	9/7/1	8/1	9/8/1												
1011	7/2	9/7/2	8/2	9/8/2												
1100	7/3	9/7/3	8/3	9/8/3												
1101	7/4	9/7/4	8/4	9/8/4												
1110	7/5	9/7/5	8/5	9/8/5												
1111	7/6	9/7/6	8/6	9/8/6												

Il sistema condensatore può inoltre essere fornito del dispositivo di by-pass.

Volture

ELEA 4.115

– Manuale di programmazione –

Errata corrige e aggiornamenti al 20/4/65

Pag. 2 La 3ª e 4ª riga dal fondo diventano:

– istruzioni logiche (AND, OR e OR esclusivo) per manipolazioni sul singolo bit;

Pag. 7 La 2ª frase del 2° capoverso diventa:

Gli indirizzi sono espressi in codice binario puro mediante 13 bits, e variano perciò da 0 a 8191 occupando complessivamente due byte in memoria.

Pag. 23 Fra la 5^{a} e la 6^{a} riga dal fondo inserire:

* La rappresentazione esadecimale è impiegata per rappresentare tutte le possibili configurazioni binarie di un byte, per mezzo di coppie di caratteri di cui il primo rappresenta il semi-byte più significativo ed il secondo il semi-byte meno significativo. I caratteri componenti l'insieme esadecimale sono i seguenti:

0 = 0000	A = 1010
1 = 0001	B = 1011
2 = 0010	C = 1100
3 = 0011	D = 1101
4 = 0100	E = 1110
5 = 0101	F = 1111
6 = 0110	
7 = 0111	
8 = 1000	
9 = 1001	

Pag. 89 Prima dell'inizio del paragrafo 6.2.1 inserire:

Per i primi due tipi I1 rinvia ad un campo avente la seguente struttura:

Z	X

mentre la struttura del campo, indicato da I1, relativo al trasferimento è:

Z	X	L	I

Significati notevoli di F' di SAC

	Bit di F'				Esadec. codific.
	7	6	5	4	
- SNO = Salto mai verificato	0	0	0	0	00
- SAI = Salto incondizionato	1	1	1	1	F0
Esame di RIPO e DIVE dopo un CFN o CFR					
- SMG = Salta se maggiore	0	0	0	1	10
- SMN = Salta se minore	1	1	0	0	C0
- SEG = Salta se uguale	0	0	1	0	20
- SMGU = Salta se maggiore o uguale	0	0	1	1	30
- SMNU = Salta se minore o uguale	1	1	1	0	E0
- SDV = Salta se diverso	1	1	0	1	D0
Altri esami					
- Salta se risultato uguale a zero (dopo EDT, ORE, TFN)	1	0	1	0	A0
- Salta se overflow (dopo ADD o ADB)	0	0	1	1	30
- Salta se risultato in complemento (dopo SOB o SOD)	0	0	1	1	30
- Salta se carattere trovato (dopo RCA o RCI)	0	1	0	1	50
- Salta se condizione esterna verificata (dopo EST di esame)	0	0	0	1	10
- Salta se condizione esterna non verificata (dopo EST di esame)	0	0	1	0	20
- Salta se fine trasmissione su lunghezza (dopo EST di trasmissione dati su canale 1)	0	0	1	0	20
- Salta se si è avuto fine da supporto (dopo EST di trasmissione dati su canale 1)	0	0	0	1	10
- Salta se risultato uguale a zero (dopo ADD o ADB)	1	0	0	0	80
- Salta se risultato uguale a zero (dopo SOD o SOB)	0	0	1	0	20

Pag. 93 Fra la 6ª e la 7ª riga dal fondo inserire:

Essa è rappresentabile con 10 bit per memorie fino a 4096 posizioni e con 12 bit per memorie fino a 8192 posizioni.

Le righe 4÷6 dal fondo diventano:

Sono perciò indicabili lunghezze da 1 (L=0000 0000 0000 0000) a 1024 (L=0000 0011 1111 1111) caratteri per memorie fino a 4096 posizioni; da 1 fino a 4096 (L=0000 1111 1111 1111) caratteri per memorie con 8192 posizioni.

Pag. 94 Nella colonna più a destra della tabella leggere:

0 - canale 1
1 - canale 2

Pag. 95 Il 2° periodo del 2° capoverso diventa:

Sul canale 1 possono essere eseguite le operazioni di lettura dal lettore seriale integrato o di trasferimento in ingresso da, o in uscita verso, una delle unità con governo standard (compresa la stampante seriale collegabile, tramite governo standard, al connettore 1).

Pag. 96 Il periodo riportato nelle righe 8, 9 e 10 diventa:

La lunghezza indicata nel campo 1 si riferisce al numero di byte di memoria impegnati.

Pag. 97 Nella 2ª riga la frase riportata fra parentesi va sostituita con:

(ъ)

Pag. 97 Nella 4^{a} riga leggere "canale 2" in luogo di "canale 1".

Pag. 102 Nella colonna che porta l'intestazione "Lettore",in corrispondenza della riga "FISE", scrivere:

Fine sequenza

Pag. 109 Nella tabella i valori posti all'incrocio della colonna 2 con le righe SAR e SAC devono essere 0 anzichè 1.

Pag. 110 Nella riga EDT in corrispondenza della colonna 10 deve essere scritto SI anzichè NO, in corrispondenza della colonna 11 deve essere scritto NO anzichè SI.

Pag. 115 La tabella aggiornata sul "Codice intermedio compatto" è riportata nella pagina seguente.

N.B. Nelle pagine XIII, XIV e XV sono riportati i valori di Z e X relativi ai comandi interessanti le seguenti unità esterne: Lettore integrato LS 300/A, UP150 Unità di perforazione, Stampante integrata MZ4 R.

TAVOLA PER TRASCODIFICAZIONE

Codice intermedio compatto

dcba \ hgfe	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000		7	8	11												
0001	6	7/6	8/6	11/9						11/6						
0010	5	7/11	8/5	7/5						11/5						
0011	4	7/4	8/4	11/9/7						11/4						
0100	3	7/3	8/3	11/8						11/3						
0101	2	7/2	8/2	11/9/8						11/2						
0110	1	7/1	8/1	11/1												
0111	0	7/0	8/0	11/0												
1000	9	9/7	9/8	12				11/12								
1001	9/6	9/7/6	9/8/6	12/9			11/12/6	11/12/9		12/6						
1010	9/5	9/7/5	9/8/5	12/7			11/12/5	11/12/7		12/5						
1011	9/4	9/7/4	9/8/4	12/9/7			11/12/4			12/4						
1100	9/3	9/7/3	9/8/3	12/8			11/12/3	11/12/8		12/3						
1101	9/2	9/7/2	9/8/2	12/9/8			11/12/2			12/2						
1110	9/1	9/7/1	9/8/1	12/1				11/12/1								
1111	9/0	9/7/0	9/8/0	12/0				11/12/0								

TAVOLA PER TRASCODIFICA codice BULL

LETTORE INTEGRATO LS 300/A

N° UNITA' = 80

		Z		X
		Can. 1	Can. 2	
COMANDI E TRASFERIMENTI				
Lettura esecutiva normale	*	10	11	40
Lettura esecutiva con ricombinazione	*	00	-	40
Predisp. Lettura normale	*	-	19	40
Selezione 2^ Casella	*	80	81	48
Poni by-pass	*	80	81	A0
Togli by-pass	*	80	81	A1
Reset errore	*	80	81	47
ESAME CONDIZIONI				
Test errore	*	C0	C1	03
Test fine schede (fine sequenza)	*	C0	C1	05
Test fuori servizio	*	C0	C1	14
Test caselle piene	*	C0	C1	1E
Or dei 4 test precedenti	*	C0	C1	44
Test pronto alla lettura (unità disponibile)		C4	C5	01
Test pronto alla alimentazione		C4	C5	10

* = Comando o Esame condizione con attesa di unità disponibile.

N.B. = I valori di Z e di X sono rappresentati in codice esadecimale codificato (0÷9, A÷F).

UP 150 UNITA' DI PERFORAZIONE
con governo standard

N° UNITA' = 00 (nella composizione base) o 64

		Z	X
		Can. 1	
COMANDI E TRASFERIMENTI			
Avvio unità meccanica (START MOTORE)		84	11
Arresto unità meccanica (STOP MOTORE)	*	80	14
Riemp.buffer, aliment. e perforazione	*	50	42
Alimentazione scheda	*	80	0C
Selezione 2^ Casella	*	80	48
Reset errore	*	80	47
Poni by-pass	*	80	A0
Togli by-pass	*	80	A1
ESAME CONDIZIONI			
Test errore	*	C0	03
Test fine schede (magazzino vuoto)	*	C0	12
Test fuori servizio	*	C0	14
Test caselle piene	*	C0	1E
Or dei 4 TEST precedenti	*	C0	44
Test governo pronto		C4	01
Test motore in funzione e tensioni continue (macchina attivata)		C4	1A

* = Comandi o esami con attesa di disponibilità.

N.B. = I valori di Z e di X sono rappresentati in codice esadecimale codificato (0÷9, A÷F).

STAMPANTE INTEGRATA MZ4 R

N° UNITA' = CO

		Z	X
		Can. 2	
COMANDI E TRASFERIMENTI			
Predisposizione stampa	*	59	42
Stampa normale	*	51	42
Stampa caratteri magnetici	*	51	43
Interlinea semplice	*	81	0A
Interlinea doppia	*	81	59
Salto su Can. 1	*	81	51
Salto su Can. 2	*	81	52
Salto su Can. 7	*	81	57
ESAME CONDIZIONI			
Test fine carta	*	C1	12
Test fuori servizio	*	C1	14
Test ERAR (ERR2)	*	C1	42
OR dei 3 Test precedenti	*	C1	44
Test stampante libera (unità disponibile)		C5	01

* = Comando od esame con attesa di unità disponibile.

N.B. = I valori di Z ed X sono rappresentati in codice esadecimale codificato (0÷9, A÷F).